An Interop Publication

ConneXions

The Interoperability Report

June 1992 Volume 6, No. 6

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

Components of OSI: The RPC Service.................... 2

IETF Internet Audiocast...... 10

INTEROP 91 Fall shownet...18

Status of SNMP Evolution... 22

SLIP Interoperability.......... 24

OSI is Still a Good Idea........ 33

Letters to the Editor............ 36

Book Review......................... 38

Announcements....................39

ConneXions is published monthly by Interop Company, 480 San Antonio Road, Suite 100, Mountain View, CA 94040, USA. 415-941-3399. Fax: 415-949-1779. Toll-free: 1-800-INTEROP.

E-mail: connexions@interop.com.

ISSN 0894–5926

From the Editor

With INTEROP 92 Spring just over, it is time to get the June issue out the door, so the report from the show will have to wait for a while. The only INTEROP related article you'll find in this issue is a look back to last fall: Brent Chapman gives us an insider's look at the building and management of the show network (page 18).

First, another installment in our *Components of OSI* series. This time we examine the standard for the Remote Procedure Call Service, currently under ballot for progression to Draft International Standard (DIS). The article is by Alan Turner from Battelle's Pacific Northwest Laboratory.

For the last year or so, we've been running articles about "neat new stuff"—applications developed for use in the Internet (or in private internets) that go far beyond the basic functionality of e-mail, file transfer and remote login. A very recent addition to this collection of new applications was demonstrated at the San Diego IETF meeting earlier this year. Live audio from the IETF site was "audiocast" using IP multicast packet audio over the Internet to participants at 20 sites on three continents spanning 16 timezones. Steve Casner and Steve Deering describe the setup in an article on page 10. We have more articles coming on new Internet applications. Stay tuned for *Prospero, Gopher* and *World Wide Web* in future issues.

SNMP has clearly become the network management protocol of choice in the industry. James R. Davin, the Area Director for network management in the IETF, describes how the Internet management framework allows for an evolution, and how you can take part in that process.

While *ConneXions* does not normally engage in product comparison or review, it *is* an "interoperability report." Therefore, when I came across the *Serial Line Internet Protocol* (SLIP) interoperability study performed by Richard Coop from the University of Hull, I decided to ask for an article. The text illustrates some very important lessons about the interworking between different implementations of the same basic protocol, and is based upon research performed for the UK Internet Consortium (UKIC). The results were first presented at a "mini-INTEROP" hosted by UKIC in February of this year.

The Great OSI Debate drew a large crowd at INTEROP 92 Spring, and reactions to it and my opinion piece in the May issue is starting to arrive. This month we bring you the updated version of Dick desJardins' argument, as well as a few letters to the Editor on the same topic. Keep your letters coming on this and any other subject.

Another *Special Issue* of *ConneXions* on electronic mail and directory services is taking shape and will most likely be released in August.

Components of OSI: The Remote Procedure Call (RPC) Service

by Alan E. Turner II, Battelle—Pacific Northwest Laboratory

Introduction

Distributed computing using the client–server model has become increasingly common. Client–server computing distributes the processing of an application between two or more systems whose interaction is coordinated with function-specific protocols.

Remote Procedure Call (RPC) is a general purpose mechanism to support distributed computing [1]. Rather than supporting a specific function (e.g., window systems, database access), RPC systems allow the development of distributed applications which support whatever functions their designers envision.

RPC systems operate by extending local procedure calls to operate between processes on separate networked systems. The effect is to hide the complexity of operating in a network environment (e.g., protocols, error recovery, data representation, addressing) from the application developer and replace it with the illusion that a distributed system is local. The client and server interact using what appear to be local procedure calls.

Several widely used commercial systems that use RPC are currently on the market (e.g., Netwise's *RPCtool,* OSF's *DCE,* and Sun's *ONC*).

Existing OSI protocols offer most of the support needed to provide an RPC service. ISODE, a publicly available implementation of OSI, contains an "application cookbook" [2] which delivers RPC capabilities by building on the available OSI services. However, if different implementations of RPC were to interoperate a new application layer standard was required.

The RPC Standard

This article describes the OSI RPC as specified in the Committee Draft standard [3] currently under ballot for progression to *Draft International Standard* (DIS). The article takes a brief tour through the standard, outlining its purpose, describing many of its key features, and providing examples of its usage.

The OSI RPC standard has been written to ensure the future interoperability of distributed applications that use RPC. It is also hoped that providing a general purpose mechanism in OSI will reduce the need for application-specific protocols.

RPC is unusual among OSI application layer standards in two significant ways. It is intended to be used directly by application programmers, and it does not specify a complete protocol but rather a basic protocol framework, a language (the interface definition notation) for specifying interfaces, and rules to convert a specific interface to a complete protocol.

Interaction Model

The interaction model defines the style of interactions supported by OSI RPC by relating its concepts (including interface, binding, calls, context, terminations, and cancels) to those of programming environments. These concepts are discussed in the following paragraphs. However, the model should be interpreted at an abstract level and is independent of any particular implementation technique, programming language, or communications protocol.

Interactions are defined without reference to, and are independent of: The location of the procedures; the data communications services used; the programming languages used to produce the procedures; the data representation used by programming languages or systems.

Interfaces and bindings

OSI RPC groups a set of procedures into an *interface*. The interface specifies the "signature" of each procedure: the procedure name, the number, types and directions of its parameters, and the terminations and types returned.

An *interface instance* is one instance of a server, offering an interface. Clients and servers are bound together at execution-time, the OSI RPC identifies each particular binding using what is traditionally called a *binding handle*. Each server may be bound to many clients and each client may also be bound to many servers. Calls to the same remote procedure using the same binding will always result in execution of the same physical procedure. A binding's lifetime is application-dependent.

A binding must exist before any remote procedure calls can occur. The interface definition contains an interface identifier which can be used by the client to locate and bind with a server.

Call semantics

All OSI RPC interactions consist of calling and executing procedures. Every call always consists of the following sequence of events (which may be incomplete if a failure or cancel occurs):

- The call occurring at the calling procedure;
- The call arriving at the called procedure;
- Execution of the called procedure;
- The return occurring at the called procedure;
- The return arriving at the calling procedure.

Multiple OSI RPC interactions can occur at the same time. The execution of a called procedure might include another remote procedure call back to its caller (a *call-back*). At any given time, there may be multiple concurrent procedure calls outstanding. The concurrency and synchronization mechanisms used in applications is not specified or restricted by the RPC standard. Systems which support parallelism (e.g., with fork and join or light weight threads) will not be restricted in their use of the OSI RPC service.

OSI RPC supports "at-most-once" call semantics. In the case of a successful return, the caller knows that the remote procedure has been executed exactly once; however, in the case of a failure return, the calling procedure may not be able to determine whether the remote procedure was executed.

It is possible for an application to combine the services of RPC and OSI transaction processing [4] to achieve exactly-once semantics. The application can then ensure that a remote procedure has been executed exactly once or never with no possibility of partial execution.

All information communicated between the calling and called procedures is explicitly contained in parameters.

Persistent context

The client and server may have relationships whose durations are longer than a single call. These relationships and their duration are application-specific. In these cases, the server is aware of the client and may explicitly maintain execution context (e.g., state information) on its behalf.

Applications typically use *context handles* to identify the context maintained between calls. A handle is generated by the application to identify some part of the context; its meaning and lifetime are determined by the application. Handles may be passed between a client and server as parameters of RPCs.

continued on next page

The Remote Procedure Call *(continued)*

For a distributed application to use context handles, the client must be able to access the same server across successive calls. The OSI RPC binding mechanism allows this.

The OSI RPC standard does not manage context handles. However, it is possible for vendors to use the extension mechanism to identify parameters that represent context information and then manage them through local means.

Terminations

A *termination* is the response generated by a particular call. One termination that is always possible is the "normal" termination—the successful completing of a procedure and return of its parameters. Since remote procedure calls are subject to sources of failure not present in the local case, additional terminations have been defined. OSI RPC supports three classes of terminations:

- *Application-specific* terminations are defined in the interface definition. This includes both the "normal" termination and any additional terminations which are specified.
- *RPC-specific* terminations are defined in the standard. These address problems such as parameter errors, insufficient resources, cancellation of the remote procedure call, and network errors.
- *System-specific* terminations are defined elsewhere, for example by system implementors.

Cancel

OSI RPC includes a *cancel* mechanism which can be used to request the cancellation of an outstanding remote procedure call. If a procedure call is outstanding when a local cancel (e.g., a Control-C) is received by the caller, the cancel can be propagated to the callee.

Interface Definitions

The OSI RPC's *Interface Definition Notation* is the language used to write an interface definition, which describes the interface between a client and a server. Since the interface definition is the user's interface to the RPC services it has been designed to be familiar to application programmers, easy to use, and as expressive as most procedural programming languages.

Interface definitions can be exchanged between systems and processed for use in supporting a distributed application. The standard does not constrain how an implementation should process an interface definition. Traditionally however, interface definitions are processed by "stub compilers" which produce code (that appears to the application to be the procedures specified in the interface definition) which is linked with the client and server portions of the application. Interface definitions can import constant, type, and procedure definitions from another interface definition to facilitate modular development.

An interface definition will generally contain several procedure declarations; some of these procedures are located on the client and others on the server. Each procedure declaration contains a list of parameters and their types and attributes, an optional return parameter, and an optional list of parameterized exceptions. Parameters each have a direction (in, out, or inout) which determines if the parameter must be transmitted on call, return, or both.

An interface definition can contain primitive data types (integer, real, character, boolean, enumerated, procedure, and octet), generated data types (record, choice, array, and typed pointer), and combinations of the two (e.g., a pointer to procedure or an array of records). In addition, the interface definition may define a data type and then use its name as a new type.

Example

The sample interface definition below illustrates some of the expressive power of the interface definition notation. For the purpose of illustration, the identification of interfaces has been simplified to a simple label, in this case "Example."

The import mechanism is used to obtain two symbols, "type1" and "type2," from the interface "External." "External" (not shown here) defines these symbols as the names of types.

The first type definition defines a record type named "rec1." This record contains an integer and a dynamic array whose size will be determined at call time. The second type definition defines an enumerated type named "enum1."

Two constant values ("startint" and "endint") are defined next.

Procedure "foo," which is located on the server, has four parameters. Parameter "c" is an example of a multi-dimensional array type with dynamic bounds in the first dimension "a..b." The actual bounds are determined at the time of the call by examining parameters "a" and "b." Parameter "d" is an example of a typed pointer.

Procedure "bar," which is located on the client, has three parameters and allows a return value. Parameter "h" is an example of the choice type. Choice types always include a discriminant, in this case "g," whose value is used at call time to determine which alternative field is actually in use ("p" if g is between 1 and 5, "q" if g is between 6 and 9, or "r" otherwise).

```
interface "Example"
  begin
    imports type1, type2 from "External";
    type rec1 = record of (flda: integer,
                 fldb: array(*) of character );
    type enum1 = enumerated ( red, blue, green );
    value startint : integer = 10;
    value endint : integer = 20;
    procedure foo    (
     in     a: integer,
     in     b: integer,
     inout c: array( a..b ,startint..endint ) of type1,
     in     d: pointer to rec1
     );
    [ client ] procedure bar (
     out    f: type2,
     in     g: integer,
     in     h: choice ( g ) of (
                select ( 1..5 )p: real,
                select ( 6..9 )q: boolean,
                default         r: enum1
                     )
     ) returns( s: real );
  end
```

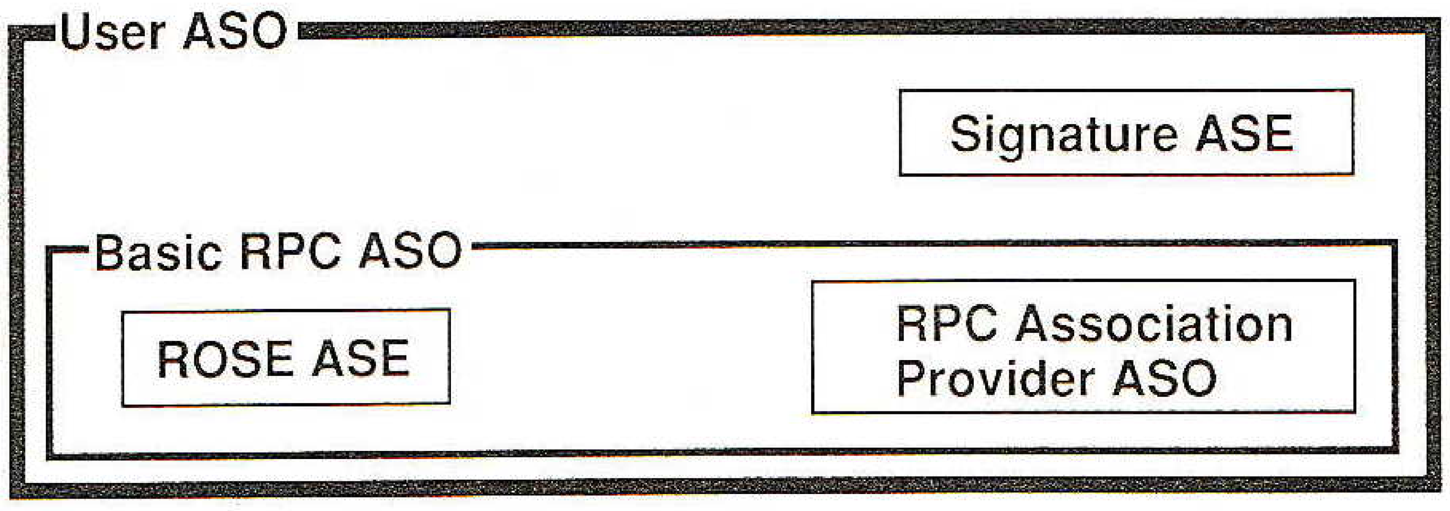
Communication Model Service and Protocol

An earlier article in this series [5] described the OSI *Application Layer Structure* (ALS) and *Application Service Elements* (ASEs). The recently extended ALS (or XALS) [6] provides a recursive structure which supports the inheritance of services within the application layer. An XALS *Application Service Object* (ASO) may contain a set of ASEs or ASOs or both. OSI RPC is the first standard to utilize XALS concepts in its specification.

continued on next page

The Remote Procedure Call *(continued)*

The OSI RPC standard uses five application layer components to perform remote procedure calls. These components are the *User ASO*, the *Signature ASE*, the *Basic RPC ASO*, the *ROSE ASE*, and the *RPC Association Provider ASO*. These elements are configured like this:



Note that OSI models are not intended to represent implementation architectures and are never subject to conformance.

The User ASO is the component directly accessible to the application. It provides a "custom" set of services which appear to be the procedures defined in the interface definition. Its services include invoking specific procedures, returning their results, and reporting errors. In addition it provides the RPC cancel services to the application. The primary function of this ASO is to coordinate the operation of the Signature ASE and the Basic RPC ASO.

The Signature ASE is "custom" built for a specific interface definition by applying rules contained in the standard. The Signature ASE understands each procedure and its parameters and offers services to invoke, return, and report errors. The ASE produces an "operation number" and "argument data" for each invoke.

The Basic RPC ASO provides RPC services which do not vary with the application. This ASO views procedures as being numbered "operations" which each take a single chunk of "argument data." The services provided are to invoke operations, return operation results, cancel operations, and to report and manage errors.

The ROSE ASE [7] produces simple protocol to request a remote operation and return the operation's results or errors. The Basic RPC ASO uses ROSE to package its "operation numbers" and "argument data" into protocol for the invokes and returns.

The RPC association provider ASO provides communications to transfer information between client and server. The Basic RPC ASO uses this ASO to transfer the protocol generated by ROSE.

Use of Associations

OSI application layer entities communicate over "associations." The RPC association provider ASO (AP-ASO) controls how these associations are established, used, and released.

Designers of the OSI RPC standard considered many alternatives for managing associations:

- Using connection-oriented (CO-mode) associations whose duration would be a single call.
- Using, or sharing, connection-oriented associations whose duration might be several calls.
- Using connection-less (CL-mode) associations (whose durations by definition are always a single message).

Each alternative has differing strengths and weaknesses; each could be the "best" given the right application and environment. In fact, all three schemes have been used by existing RPC systems.

It is possible to implement any of the schemes by designing an appropriate AP-ASO. However, RPC implementations which use different AP-ASOs may not interoperate.

The standard currently specifies an AP-ASO which manages CO-mode associations with a duration of one call. The service is a "one-shot" which bundles together the association establishment, call, return, and association release. This allows implementations to approach the CL-mode style in performance for simple client-server relationships while avoiding the complexity of coordinating the CL-mode associations.

The standard also contains an optional alternate AP-ASO which manages CO-mode associations with a long duration. This ASO would offer better performance for applications which perform "bursts" of procedure calls or which use other OSI services which also maintain associations.

Data Representation

Since OSI RPC is an application layer standard it is only indirectly concerned with data representation. These concerns are handled by the presentation layer and through the use of *Abstract Syntax Notation One* (ASN.1), both summarized in earlier articles in this series [8, 9].

It is the responsibility of the presentation layer to "negotiate" the actual data representations used to exchange information. To simplify, all data is described using the abstract notation and represented by applying particular encoding rules to the local data so described.

There has been concern that any RPC which uses the OSI *Basic Encoding Rules* (BER) will be inefficient. The newer OSI encoding rules (e.g., light weight encoding rules, packed encoding rules) can be implemented more efficiently than BER and will no doubt be used to support RPC.

Other encoding rules are possible (and can be defined by groups other than ISO/CCITT) which can handle the data representation problem in whatever way their designers desire, including every method in use by current RPC systems.

Future

Two non-OSI international standard projects, together known as the *Common Language Independent* (CLI) work, are addressing the problems of language independent procedure calls [10] and data types [11]. CLI will enable conforming programming languages to participate in a multi-language environment by being able to consistently call each other and understand the same data types.

The procedure call model and data types of CLI and OSI RPC are identical. As language groups complete their procedure call and data type mappings for CLI, all of that work will be directly applicable to OSI RPC; this will save considerable time and expense.

Using RPC and CLI it will be possible for application developers to call a routine without knowing what language the routine was written in, what system the routine executes on, or even if the routine is local or remote.

Conclusion

Although the OSI RPC standard is complex it is also designed to be easy to use by application programmers and to be efficiently implementable. When the standard reaches Draft International Standard (DIS) level it will be technically stable, a key factor in implementation decisions. As with most standards, the market will then decide what happens next.

The Remote Procedure Call *(continued)*

By providing the application programmer with a sophisticated tool, OSI RPC reduces the difficulty of creating distributed applications. A significant barrier to network usage (programming complexity) is reduced; the need to create and support more protocols is decreased; and most significantly, the opportunity to economically realize new services tailored to specific problems is increased.

References

[1] Bruce Nelson, "Remote Procedure Call," Ph.D. thesis, Carnegie-Mellon University, Computer Science Department, 1981.

[2] Marshall Rose, "Building Distributed Applications in an OSI framework," presentation available with ISODE, 1988. Also in *ConneXions—The Interoperability Report,* Volume 2, No. 3, March 1988.

[3] International Organization for Standardization, "Opens Systems Interconnection—Remote Procedure Call Specification, Parts 1–4," ISO/IEC CD 11578.

[4] International Organization for Standardization, "Open Systems Interconnection—Distributed Transaction Processing," DIS 10026.

[5] Marshall Rose, "Components of OSI: The Application Layer Structure," *ConneXions,* Volume 4, No. 1, January 1990.

[6] International Organization for Standardization, "Open Systems Interconnection—Application Layer Structure, Amendment 1: Extended Application Layer Structure," ISO/IEC 9545/DAM 1.

[7] International Organization for Standardization, "Text Communication—Remote Operations," International Standard ISO 9072.

[8] David Chappell, "Components of OSI: The Presentation Layer," *ConneXions,* Volume 3, No. 11, November 1989.

[9] Howard Motteler and Deepinder Sidhu, "Components of OSI: Abstract Syntax Notation One," *ConneXions,* Volume 6, No. 1, January 1992.

[10] International Organization for Standardization, "Common Language-Independent Procedure Calling Mechanism, Working Draft 4," ISO/SC22/WG11 N295.

[11] International Organization for Standardization, "Common Language-Independent Datatypes, Working Draft 5," ISO/SC22/WG11 N233.

ALAN TURNER is a Senior Research Scientist in the Computer and Network Technology Department at the Pacific Northwest Laboratory, a multi-discipline research laboratory operated by Battelle for the U.S. Department of Energy under Contract DE-AC06-76RLO. Mr. Turner's current work involves the implementation of OSI services, GOSIP conformance, and related coexistence and transition issues. He is a member of the ANSC X3T5.5 (OSI Upper Layers) standards development group and an author of the OSI RPC standard. Alan holds an MS in Computer Sciences from the University of Missouri at Rolla.

"Components of OSI" in *ConneXions*

In an attempt to explain the *Open Systems Interconnection* (OSI) model, promulgated by the International Organization for Standardization (ISO) and the Consultative Committee for Telephone and Telegraph (CCITT), *ConneXions* has been running a number of articles under the heading "Components of OSI." Here is a list of them:

Integrated Services Digital Network (ISDN)	April	1989
X.400 Message Handling System	May	1989
X.500 Directory Services	June	1989
The Transport Layer	July	1989
Routing overview	August	1989
IS–IS Intra-Domain Routing	August	1989
ES–IS Routing	August	1989
The Session Service	September	1989
Connectionless Network Protocol (CLNP)	October	1989
The Presentation Layer	November	1989
A taxonomy of the players	December	1989
The Application Layer Structure	January	1990
File Transfer, Access, and Management (FTAM)	April	1990
The Security Architecture	August	1990
Group Communication	September	1990
X.25—the Network, Data Link, & Physical Layers	December	1990
The Virtual Terminal ASE	January	1991
Systems Management	April	1991
CO/CL Interworking	May	1991
Open/Office Document Architecture (ODA)	August	1991
Abstract Syntax Notation One (ASN.1)	January	1992
Broadband ISDN	April	1992
Synchronous Optical Network (SONET)	April	1992
Asynchronous Transfer Mode (ATM)	April	1992
Inter-Domain Routing Protocol (IDRP)	May	1992
The Remote Procedure Call (RPC) Service	June	1992
OSI Conformance Testing	*Coming soon*	

Also note that FDDI, while not published under the same heading, is also an ISO protocol and should be considered part of the set. You'll find articles on FDDI in the October 1990, September 1991, and October 1991 issues. There are still more articles to come in this series, so stay tuned! All back issue are available for purchase. To order a back issue, or for more information contact:

ConneXions—The Interoperability Report
480 San Antonio Road, Suite 100
Mountain View, CA 94040–1219
USA
Phone: +1 415-941-3399 or 1-800-INTEROP (Toll-free in the USA)
Fax: +1 415-949-1779
E-mail: connexions@interop.com

First IETF Internet Audiocast

by Steves Casner and Deering

Introduction

The March *Internet Engineering Task Force* (IETF) meeting in San Diego was an exciting one for those interested in teleconferencing. In addition to several sessions on teleconferencing topics, we managed to pull off a "wild idea" suggested by Allison Mankin from MITRE: live audio from the IETF site was "audiocast" using IP multicast packet audio over the Internet to participants at 20 sites on three continents spanning 16 timezones.

The audiocast included all the general sessions plus one session of the Teleconferencing Architecture BOF and the three sessions of the Audio/Video Transport working group. Unlike a radio broadcast, the remote participants could talk back as well, as demonstrated during a brief technical presentation on the experiment. Though there were some problems, we knew this experiment was a success when, during the working group sessions, remote participants were able to ask cogent questions and engage in the discussion.

This event was a pilot experiment that we hope will be expanded at future IETF meetings to reach more destinations and to include video, images and "shared whiteboards" along with audio. This is a step toward a more distributed IETF, a goal Dave Farber and Jack Haverty challenged the IETF community to pursue during a discussion on the IETF mailing list last fall. This was also a demonstration of technology developed and tested in the DARTnet research testbed network sponsored by DARPA.

What was required

Two key elements were required to put the audiocast together:

- Hardware and software to generate and receive audio packets at the endpoints.
- IP multicast routing [1] to replicate the packets efficiently for distribution to a large number of recipients.

Audio hardware is now built into many workstations, such as Sun and NeXT, and is ready for deployment of software. To simplify this pilot experiment, we kept to the same hardware and software already tested in DARTnet. Of four interoperable packet audio programs, three run on Sparcstations (*VT* from ISI, *vat* from Lawrence Berkeley Laboratory (LBL), and *NEVOT* from University of Massachusetts) and one runs on a 386 PC plus audio card (from MIT).

At IETF and most of the remote sites, we ran *vat*, the *Visual Audio Tool*, written by Van Jacobson and Steve McCanne. In an X window, *vat* displays VU meters and volume control sliders for the microphone and speaker levels plus a status display identifying the participants in the conference.

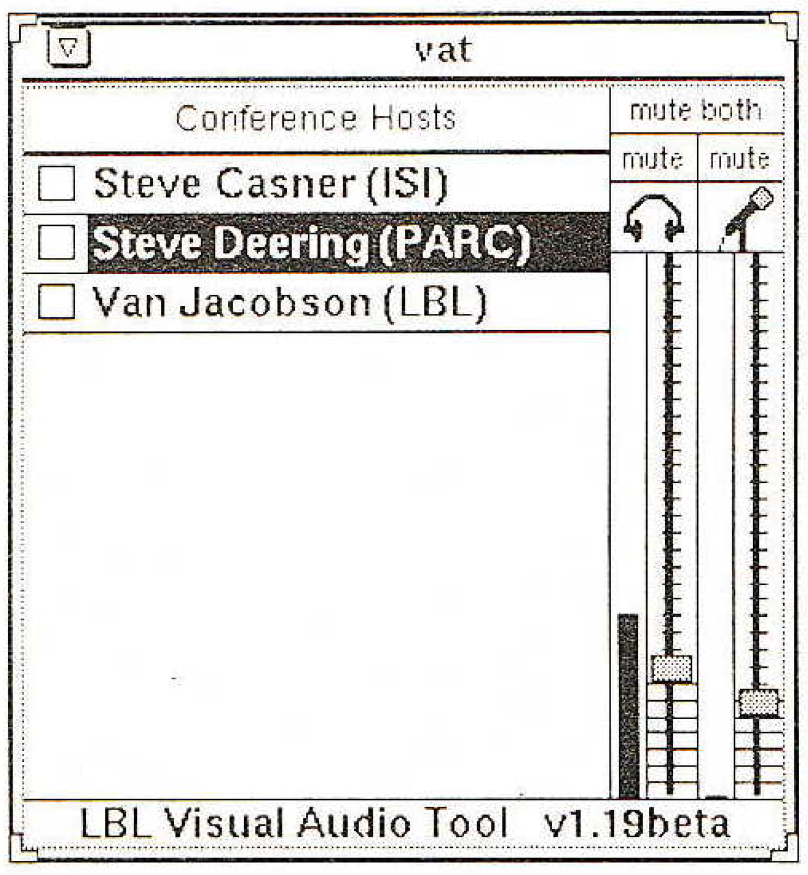


Figure 1: Image of a *vat* window

Behind the user interface, there are several important functions required to process audio packets:

- Silence/sound detection to avoid sending packets during silence, preferably with a dynamic threshold to accommodate varying levels of background noise.
- An adaptive playout delay to achieve continuous playback even though network delays vary. For multimedia systems, playout delays are also used to achieve synchronization.
- Resequencing of out-of-order packets and filtering of duplicate packets within the playout delay buffer.
- Mixing of audio sample streams when packets arrive from multiple sites at the same time.
- A means to suppress acoustic feedback from the loudspeaker to the microphone that produces an echo at the far end (headphones are one solution, half-duplex speakerphone mode is another).

Voice data rates

For this audiocast, we used the 64Kbps PCM audio produced by the Sparcstation's `/dev/audio` device directly. Each packet contains 180 PCM voice samples, corresponding to an interval of 22.5 milliseconds and a rate of 44.4 packets per second. As shown in Figure 2, the packet overhead is 32 bytes not counting any MAC header, resulting in a peak overall data rate of 75Kbps. However, since no packets are transmitted during silence, the average data rate is less. As software bandwidth compression algorithms are implemented in packet audio programs in the future, it will be possible to reduce the data rate for operation over slower network links.

IP	UDP	NVP	PCM Voice Samples
20	8	4	180

(octets)

Figure 2: Audio Packet Format

Protocols

Note that packet audio is transported by UDP rather than TCP, for two reasons. First, TCP's reliability and flow control mechanisms aren't appropriate. Occasional packet loss causes only a small and acceptable reduction in audio quality, while allowing time for retransmission would require a longer playback delay, making interactive conversation more difficult. Flow control is not required because packets are generated at a regular rate and must be communicated and consumed at that rate. (Some systems may allow the data rate to be adjusted to compensate for network load.)

The second reason for choosing UDP is that it works well with IP multicast to reach many destinations, while TCP is limited (so far, at least) to point-to-point connections.

NVP-II

UDP lacks two functions of TCP that are needed: *packet reordering* and *duplicate filtering*. We add another protocol layer to provide these functions. As an interim convention, we use the data packet header from the *Network Voice Protocol* (NVP-II) [2], as shown in Figure 3 on the next page. For this version of PCM audio, the *timestamp* field increments every 22.5 millisecond packet interval, including during silence when no packets are transmitted.

continued on next page

First IETF Internet Audiocast *(continued)*

This provides sequencing and duplicate detection within a packet lifetime of about 20 seconds. The separate sequence number increments once per packet transmitted, enabling the detection of lost packets versus packets not transmitted during silence.

Seq #	Timestamp	Chksum	OptLen
6	10	8	8

← 32 bits →

Figure 3: NVP Header Format

The NVP header is efficient (only 32 bits), but that makes some of the fields too small to support current requirements. It is in the charter of the Audio/Video Transport working group to devise one or more replacement protocols for packet audio, video and perhaps other media. At the San Diego IETF, a minimal strawman protocol was proposed and we discussed what functions should be added to it. Send a message to `rem-conf-request@es.net` to join the discussion.

IP Multicasting

Besides end-system hardware and software for packet audio, the second key element required for the IETF audiocast was IP multicasting. For a large conference like the audiocast, the bandwidth and processing required for the source host to send a separate copy of each packet to each destination would be prohibitive. With IP multicast extensions implemented in the participating hosts and routers, the source can send a single copy of each packet which is then replicated as needed at each branching point on the logical tree reaching out to the destinations.

Unfortunately, few routers in the Internet implement IP multicast routing yet. Fortunately, the experimental DARTnet routers do, using the *Distance Vector Multicast Routing Protocol* (DVMRP) implemented by Steve Deering in the *mrouted* daemon plus kernel extensions. This makes it easy for us to hold DARTnet experimenters' teleconferences every week using packet audio and video. DARTnet was also available for use as a transcontinental multicast backbone for the audiocast. But just as we need to reach some DARTnet experimenters not directly connected to DARTnet, we needed to extend past DARTnet to the audiocast sites.

Tunnels

In order to support multicasting among subnets that are separated by (unicast) routers that do not support IP multicasting, *mrouted* includes support for "tunnels," which are virtual point-to-point links between pairs of *mrouteds* located anywhere in the Internet. To transmit a multicast packet through a tunnel, a multicast router modifies the packet by appending an IP *Loose Source Route* option to the packet's IP header. The multicast destination address is moved into the source route, and the unicast address of the router at the far end of the tunnel is placed in the IP *Destination Address* field. Thus, the packet looks like a normal unicast packet to the routers and subnets along the path of the tunnel. The router at the far end of the tunnel restores the original multicast destination address and deletes the source route before forwarding the packet.

For DARTnet teleconferences, we need only a few tunnels extending from DARTnet nodes to individual sites, but for the audiocast we needed to expand this idea to a whole network of tunnel links with DARTnet as a backbone. That network, shown in Figure 4, set a record as the largest IP multicast network to date, with 34 routers linking 40 subnets.

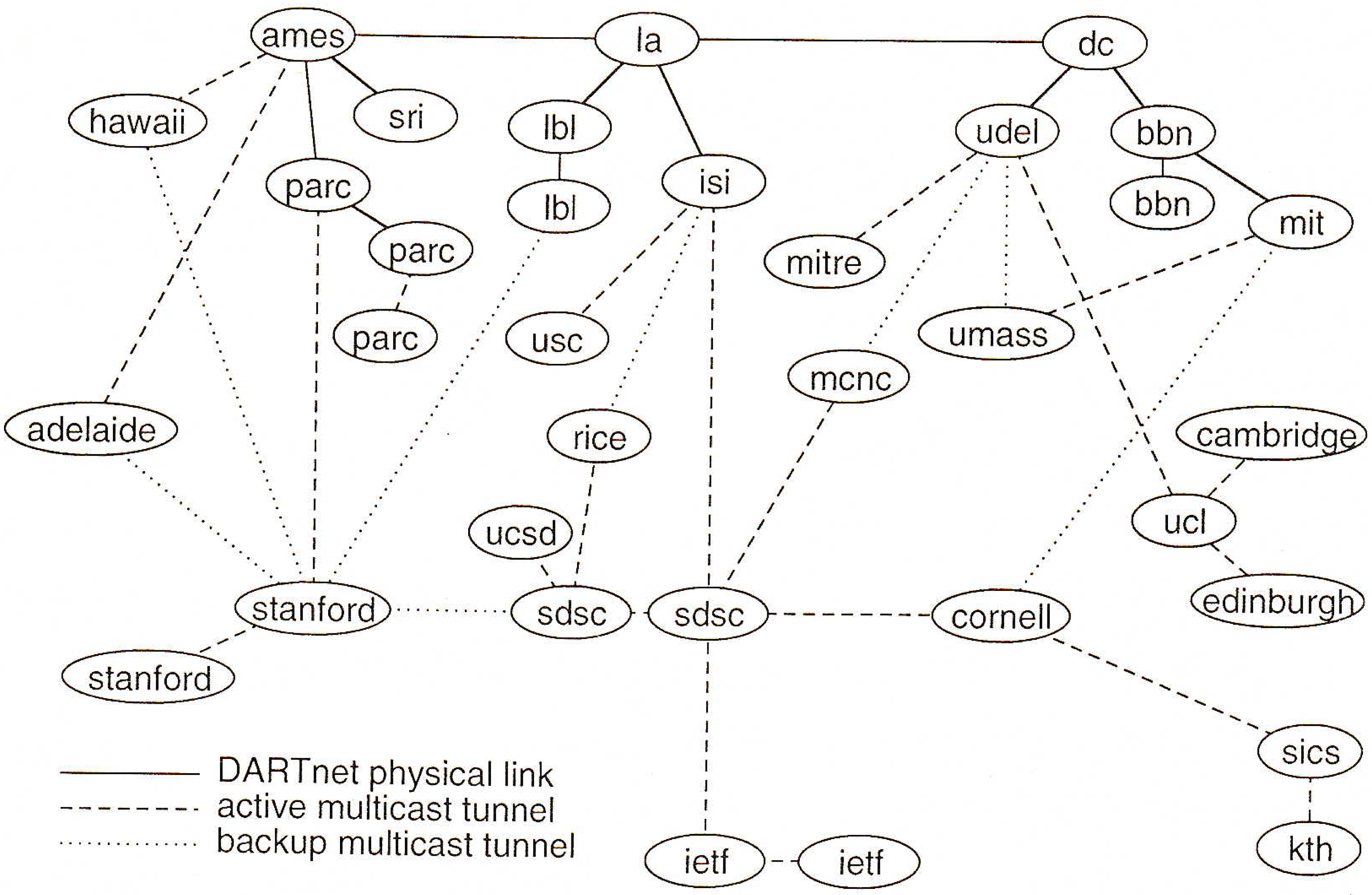


Figure 4: Multicast Router Topology for IETF Audiocast 3/92

The solid lines near the top of the graph indicate physical DARTnet links. The other lines are all multicast tunnels (virtual links) over a variety of Internet paths including the T3 NSFNET and international links. By assigning different metrics to different links, we established primary tunnels (dashed lines) and backup tunnels (dotted lines) to be used in case of failure in the primary tunnels or DARTnet lines.

What we learned

The IETF audiocast was a valuable learning experience! Van Jacobson produced five new versions of *vat* during IETF week, adding new features to improve performance. For most of the audiocast, listener reports of the sound quality ranged, over time and place, from "very clear" to "intelligible." Listeners were able to comprehend most of what was said, but there were several factors that caused dropouts in the playback.

The most obvious and expected cause of dropouts was that competing traffic would cause variance in the transit delay, especially on longer paths. To compensate, Van made improvements in *vat*'s delay adaptation algorithm and added a "lecture mode" that lets the playout delay remain large to minimize the number of packets declared late, for use during lectures when reduced interactivity is acceptable.

Audio pickup

The second problem had nothing to do with packets, but with acoustic audio pickup and silence suppression. During the Teleconferencing Architecture BOF session, we did not have a lapel microphone and attempted to pick up the presenters with a table microphone. The signal-to-noise ratio was too low because the presenters were too far from the microphone and there was too much ambient noise in a room divided by movable walls. A related problem was that, even when the speaker used a lapel microphone, comments from the audience were not picked up well enough by the room microphones and were suppressed as silence. Remote listeners heard the presenter's punch line but didn't hear the set-up comment from the audience. This was frustrating. To avoid this problem, Van added a control in *vat* to turn off silence suppression for use during presentations.

continued on next page

First IETF Internet Audiocast *(continued)*

On the other hand, disabling silence suppression may exacerbate network loading and the resulting dropouts. Simon Hackett postulated that presenting a continuous load makes a non-trivial increase in the average bandwidth and eliminates the gaps that may give routers a chance to empty their queues. In addition, the packet audio receivers generally make playout delay adjustments during silence intervals, so drift between the sending and receiving clocks must be accommodated in some other way for continuous transmission.

Packet loss

The third cause of dropouts was a persistent estimated 10–20% packet loss that we could not find nor explain. This loss was not evident in tests with *ping* and *traceroute*. Apparently there was no loss across the local T1 line connecting the source host in the main ballroom to the terminal room where another host running *vat* monitored the signal. We believe the loss occurred somewhere between the IETF terminal room and DARTnet, perhaps between IETF and the San Diego Supercomputer Center (SDSC). Even though a terminal room full of nerds banging on keyboards amounts to quite a few packets per second, that should not have created enough congestion to cause persistent, low-level packet loss.

One potential explanation would be that the multicast tunnel packets use the IP Loose Source Route option, which diverts those packets to a slower processing path in some router architectures. However, we also tested with some source-routed test traffic at rates similar to the audio and did not see the same loss. In any case, there are plans to modify *mrouted* to encapsulate in a separate IP header rather than use the source route option.

We would likely have been able to find the cause of the packet loss if we had enough time to investigate fully, but we also had real IETF business to do! Clearly we need better measurement tools and procedures to accelerate the process. It would also have been very helpful if we had some way to monitor reception at a distance so we could tell when there are problems without the users having to contact us.

Expansion of services for future IETFs

While the audiocast alone was interesting to the remote participants, in part because of its novelty, it is clear that we need to provide some of the visual content of the presentations as well. As one experiment, Ralph Droms made available via FTP the *PostScript* and ASCII versions of the transparencies for his talk on the *Dynamic Host Configuration Protocol*. The response was very positive.

Over DARTnet we regularly send packet video as well as packet audio. The data rate we typically use is only 128Kbps, twice the audio rate. It should be possible to implement software to decode that video and display it in an X window. This is under investigation at ISI. Other implementations are also feasible, for example some software encoding of video grabbed at a slow frame rate from a device such as the Sun VideoPix card.

Video of the presentations would give a better sense of what was happening, but video does not have adequate resolution to show slides very well. For that, an automated slide presentation tool would be preferable, working from on-line source material or an on-site scanner. To generalize further for interactive working group discussions as well as presentations, a "shared whiteboard" onto which slides can be posted would be even better. Several research projects are working on such tools. Future IETF meetings would provide an opportunity for a trial by fire for these tools.

Perhaps the biggest impediment to expanded services will not be network bandwidth or implementation of new tools, but logistics at the IETF site. We had enough trouble this first time trying to figure out how to connect into the ballroom audio system in such a way that a moderator could preview questions from remote sites before disturbing the local participants. Video cameras and people to operate them will add another logistical dimension the IETF Secretariat may be reluctant to tackle, not to mention the additional expense.

Yet another level of complexity would be introduced if multiple working group meetings are to be teleconferences. Especially for the larger working groups, room acoustics would be a problem, both for picking up the voices of all participants and to play the sound from the remote end at sufficient volume and still cancel the echo caused by the microphones picking up that sound. Also, the resolution of compressed video is not sufficient for more than about 6 people. Yes, rooms can be outfitted with equipment and video production personnel to handle these problems, but the cost is substantial. Meeting sites with a video-equipped auditorium and 20 breakout rooms equipped for videoconferencing are probably rare.

Network connectivity within the site is another issue. At San Diego, we were unable to get twisted pair Ethernet to reach between buildings at the hotel over some crusty old wiring passing through the damp pool filter room. We had to back off to more bulky T1 equipment, and even that started getting line errors on the fourth day.

Packet audio does need reasonable bandwidth, after all. To support several working group teleconferences may require more than a single T1 line from the IETF site to the Internet connection point. Beyond that point, the next hurdle is wide-area network support for real-time traffic.

Scaling up to widespread use

To scale up for widespread use of packet audio and video in multiple simultaneous public meetings, private teleconferences and other applications will require additional research and infrastructure engineering in several areas including network resource management, IP multicast routing and connection/session management.

Need resource management

The current work on protocols for packet audio and video transport over UDP is considered experimental because UDP transmission is only sufficient for small-scale use over fast portions of the Internet. If many people tried to send packet audio today, significant network congestion would likely result. Since packet audio does not practice congestion control, well-behaved TCP traffic would back off and let the audio take over which is probably not fair. Even so, when there is congestion, the resulting packet loss would impair the audio quality as well.

Research is underway on DARTnet and elsewhere to develop resource management (or traffic control) algorithms to solve this problem. These algorithms, running in the various levels of packet switches in the network, would give priority to real-time traffic such as audio and video to achieve low delay and packet loss. At the same time, the algorithms would prevent real-time traffic from using more than its fair share, as determined by payment or policy. On small links with a low degree of multiplexing, new calls may be blocked when there is insufficient capacity to avoid degrading interference with established calls. The audio/video transport protocols may be used in conjunction with other protocols such as ST-II [3] or connectionless resource setup protocols to access the resource management functions.

continued on next page

First IETF Internet Audiocast *(continued)*

Need real IP multicast

The tunnel mechanism allows *mrouted* to establish a virtual internet, for the purpose of multicasting only, that is independent of the physical internet, and that may span multiple administrative domains. However, this capability is only intended for experimental support of Internet multicasting, pending widespread support for multicast routing by the regular routers. *Mrouted* suffers from the well-known scaling problems of any distance-vector routing protocol, and does not (yet) support hierarchical multicast routing or interoperation with other multicast routing protocols such as MOSPF.

A big part of the effort required to set up this audiocast was in constructing the multicast virtual network by hand and testing its performance. We need multicast routing in the real networks to reduce that effort.

Need control protocols

IP multicast addressing works well for broadcast-type applications such as this audiocast where a priori agreements on addressing and media encoding can be made. A pre-defined multicast address and UDP port number are built in to *vat,* so to join in you just start listening. To expand from a single broadcast to multiple, private teleconferences, connection/session management protocols will be needed to request call acceptance, negotiate compatible encodings, dynamically allocate IP multicast addresses, etc. Since IP multicast traffic may be received by anyone, the control protocols must handle authentication and key exchange so that the audio/video data can be encrypted.

Like resource management, connection management is the subject of current research. We expect that standards-track protocols integrating transport, resource management, and connection management will be the result of later IETF working group efforts.

You too can play

Meanwhile, small-scale experiments with packet audio and video are encouraged to learn more about the protocol requirements. You can participate. A pre-release of the LBL audio tool *vat* is available by anonymous FTP from `ftp.ee.lbl.gov` in the file `vat.tar.Z`. Included are a Sun-4 binary suitable for use on any version of Sparcstation and a manual entry. The authors of *vat* say the source will be released "soon."

In addition, a beta release of both binary and source for the UMass audio tool *NEVOT* was recently made available by anonymous FTP from `gaia.cs.umass.edu` in `pub/nevot-0.9.tar.Z`.

You can test *vat* or *NEVOT* point-to-point between two hosts with a standard SunOS kernel, but to conference with multiple sites you will need a kernel with IP multicast support added. IP multicast invokes Ethernet multicast to reach hosts on the same subnet; to link multiple subnets you can set up tunnels, assuming sufficient bandwidth exists.

You don't need kernel sources to add multicast support. Pick up the file `vmtp-ip/ipmulti-sunos411.tar.Z` by anonymous FTP from host `gregorio.stanford.edu`. It contains the IP multicast code to be added to a SunOS 4.1.1 kernel.

Once you build the kernel, you should use *adb* to permanently patch the kernel variable *audio_79C30_bsize* from the standard value of 1024 to be 180 decimal to match the audio packet size for minimum delay. Otherwise there will be bad breakup when sound from two sites gets mixed for playback.

If you don't have a microphone for your Sparcstation, Sun is now selling one (part number 370–1414) or you can pick up an inexpensive microphone from Radio Shack. Walkman-style headphones are also recommended.

Acknowledgements

Many people helped to make this audiocast work. We would particularly like to thank Paul Love, Tom Hutton and the rest of the SDSC crew for help getting the hardware and software set up at IETF; Steve Coya and Megan Davies for logistics; and Van Jacobson for work on *vat* and help in searching for the packet losses. Walt Prue set up a special route for the tunnel from San Diego to DARTnet at ISI, and Milo Medin and Jeff Burgan set up a connection from DARTnet to FIX-West for a clear shot to Hawaii and Australia.

We'd also like to thank the remote *vat* participants for their help as guinea pigs during testing. Simon Hackett in Adelaide had to get up very early in the morning; Ian Wakeman and Jon Crowcroft in London and Anders Klemets and Steve Pink in Stockholm had to stay up late at night. This points out one impediment to "distributing the IETF" that will be hard to fix: too many time zones!

Finally, we'd like to thank DARPA for support of DARTnet and the experiments thereon.

References

[1] S. Deering, "Host Extensions for IP Multicasting," RFC 1112, August 1989.

[2] Cohen, D., "A Network Voice Protocol NVP-II," USC/Information Sciences Institute, April 1981.

[3] Topolcic, C. (Ed.), "Experimental Internet Stream Protocol, Version 2 (ST-II)," RFC 1190, October 1990.

[4] Deering, S., "IP Multicasting," *ConneXions,* Volume 5, No. 3, March 1991.

[5] Deering, S. & Cheriton, D., "Multicast Routing in Datagram Internetworks and Extended LANs," ACM *Transactions on Computer Systems,* Volume 8, No. 2, May 1990.

STEPHEN CASNER received his B.A. in Mathematics from Occidental College in 1973, and his M.S. in Computer Science from the University of Southern California in 1976. Since 1973, he has worked at USC's Information Sciences Institute on network protocols and systems for real-time communication over packet-switched networks, beginning with the first packet voice experiments over the ARPANET. He is currently Project Leader for Multimedia Conferencing at ISI. He also chairs the Audio/Video Transport working group of the IETF. He can be reached via e-mail as: `casner@isi.edu`.

STEPHEN DEERING received his B.Sc. (1973) and M.Sc. (1982) from the University of British Columbia, and his Ph.D. (1991) from Stanford University. He has been studying, designing and implementing computer communication protocols since 1978, including work on X.25 software for connecting to public networks, on the first implementation of the X.400 protocol suite for e-mail, and on advanced-functionality transport protocols for distributed systems. As part of his doctoral research, he developed several new routing protocols for efficient and scalable internetwork multicasting that are now being deployed in the Internet. Dr. Deering is currently a member of the research staff at the Xerox Palo Alto Research Center, where he is continuing his work on multicast protocols and applications, and on protocols to support wireless and mobile computers in the Internet. He is an active member of the IETF and the End-to-End Protocols Research Group. He can be reached as `deering@parc.xerox.com`.

Building & Managing the INTEROP 91 Fall Shownet

by Brent Chapman, Great Circle Associates

Introduction

INTEROP is a yearly networking conference and trade show that is considered the "event of the year" by many companies in the networking business. In 1992, INTEROP moves to a twice-yearly format. One of the key features of INTEROP is that all exhibitors at the show are connected to the "Shownet" so that multi-vendor interoperability is actually demonstrated, rather than simply talked about. The Shownet is planned, built, and run by volunteers from the industry.

Last year's INTEROP Fall conference, held October 7 through 11, 1991 in San Jose, drew over 32,000 attendees. The exhibits completely filled the San Jose Convention Center (all 3 large exhibit halls, the ballroom, and the concourse area in front of the exhibit halls), as well as the Diamond Pavilion across the street. When completed, the Shownet included over 300 vendors, over 400 subnets, and uncounted hundreds of hosts.

Media

The Shownet was a large, diverse network that included many types of physical media. These included unshielded twisted pair wire (UTP), shielded twisted pair wire (STP), fiber optic cable, and coax cable, as well as T1 and microwave links. When the Shownet was finished, it consisted of over 30 miles of UTP wire, over 5 miles of fiber optic cable, over 4 miles of STP wire, over 1 mile of coax, and untold thousands of modular connectors.

The underlying network consisted primarily of Ethernet, Token Ring, and FDDI, but many special "demo groups" ran demonstrations of other technologies, including Frame Relay, ISDN, and SMDS. The primary network protocols used were IP and OSI, but again, many demo groups demonstrated other technologies. OSPF was used for routing on the Shownet backbone.

Topology

The network was organized into 30 or so "ribs." In the main exhibit halls, the ballroom, and the Diamond Pavilion, ribs were UTP Ethernet and Token Ring runs for the exhibitor booths in one or two aisles. At the head of each rib was a pedestal containing the 10Base-T concentrators and router for that rib.

The concourse area consisted of three ThinNet (rather than UTP) ribs, each with a router at the end. The ribs were further grouped into four "backbones," one for each of the 3 main Convention Center exhibit halls running along the front of the exhibit halls, and one in the Diamond Pavilion across the street (the ballroom and concourse areas were attached to whichever main exhibit hall backbone happened to be closest). The rib routers in each backbone were connected via FDDI over fiber optic cable.

Each Convention Center backbone was connected via another router to the central network in the Network Operations Center (NOC), which was located in one of the "skybooth" offices that overlook the main exhibit floor. The Diamond Pavilion backbone was connected to the main network via a T3 link. A microwave link connected the terminal room in the Fairmont Hotel with the main network. The Shownet was connected to the Internet through a T1 link to BARRNet. (See Figure 1).

The builders

The group that designed, built, and managed the Shownet included a small handful of paid Interop Company employees, but was primarily made up of volunteers trading their time for the invaluable experience to be gained in working on such a network.

There were 3 main groups of volunteers; the Core Team, the Shownet Team, and the volunteer laborers. The Core Team comprised 9 individuals who were ultimately responsible for the network and who worked year-round (starting shortly after the previous year's INTEROP) to design and build it. They could be considered the "upper management" of the Shownet.

The Shownet Team consisted of 20 to 30 more people who worked primarily during the major activity periods (the cabling party in July, the hot stage in August, and the actual show in October; all discussed below) and were the "staff" of the Shownet. These could be considered the "middle management" of the Shownet.

Finally, and perhaps most importantly, there were the hundreds of volunteers who provided labor during the maximum effort periods such as the 8-hour period at the start of the show when the physical network was installed; the Shownet would not have been possible without these folks.

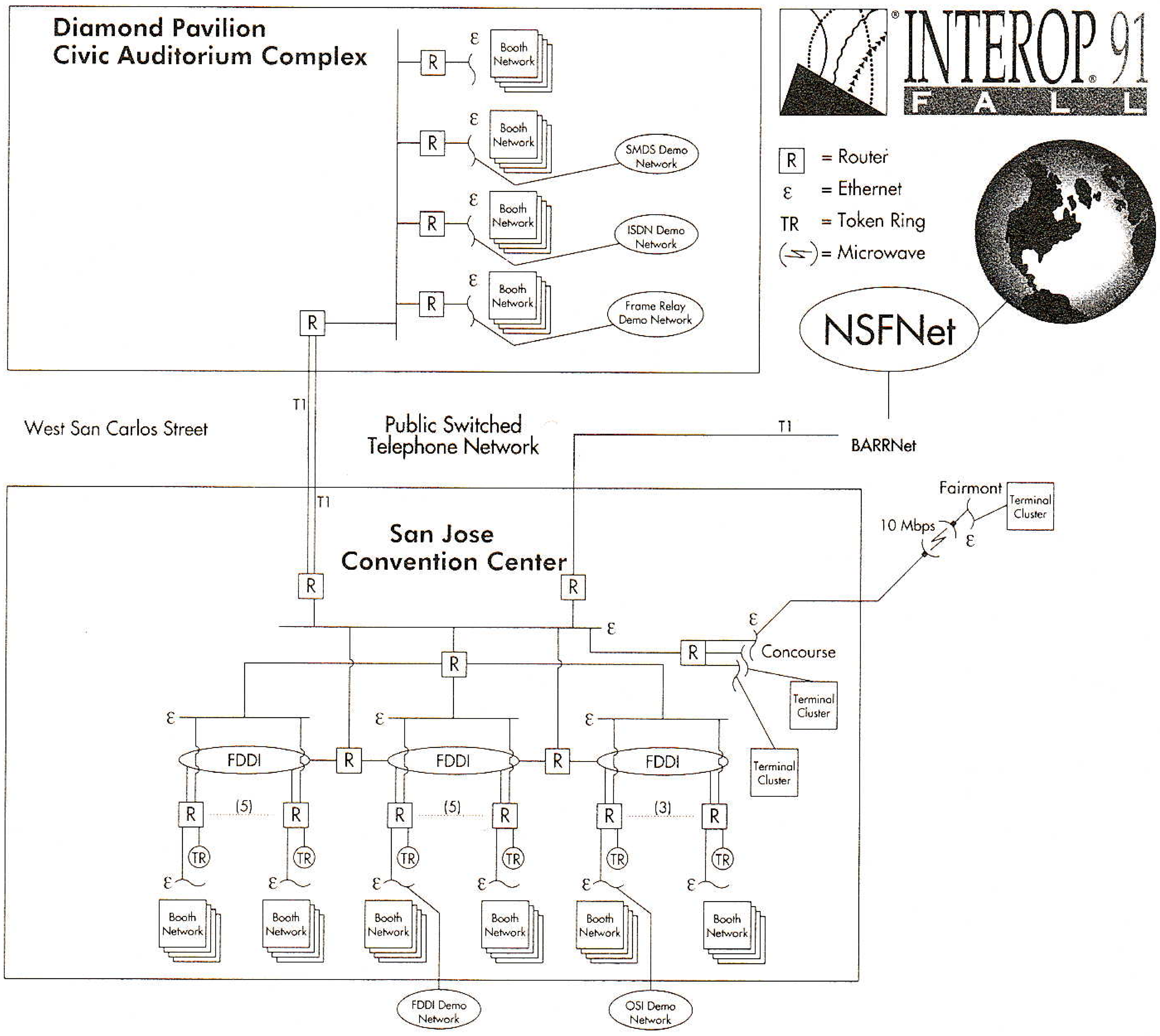


Figure 1: The INTEROP 91 Fall Shownet

Planning

The Core Team started planning the '91 Shownet shortly after the close of the previous INTEROP conference (October 1990). By late June, they had developed detailed plans for exactly what the Shownet was going to look like this year. The Core Team and the Shownet Team, plus a number of volunteers, spent the week of July 4 in the San Jose Convention Center (when there was no other conference or exhibition using the facility) laying out the physical network. The various wires were measured, cut, terminated, tested, labeled, and bundled into ribs. The ribs were rolled onto cable spools, to be stored until reassembly just before the show opened in October.

continued on next page

The INTEROP 91 Fall Shownet *(continued)*

Hot Stage

Vendors providing equipment for use in building the Shownet were required to deliver the equipment by mid-August, and in late August, the Core Team and Shownet Team gathered again, along with vendor technical representatives, for a "hot stage" event at the Interop warehouse. At the hot stage, all network equipment was assembled into the pedestals, configured, and tested. The backbone connections were configured and tested as well. The purpose of the hot stage was to discover and work out the inevitable small compatibility problems in multi-vendor installations.

In October, the weekend before the show, the Core Team, Shownet Team, and the volunteers gathered again to install and test the network. On Friday night, the Diamond Pavilion was set up, and served as something of a test case for the procedures that were going to be used to set up the Convention Center on Saturday night.

Setting up in the Convention Center was tricky, because the prior show (the Seybold Desktop Publishing conference) didn't release the show floor to Interop Company until midnight Saturday, and the network cabling that was going up to the ceiling had to be completely done by 8am Sunday, when the exhibitors would start arriving to assemble their booths.

Eight hours is not a lot of time to wire a network that size, and it would have been impossible if not for the careful planning and the several days spent in the Convention Center in July, pre-constructing all the overhead cables.

The Core Team and the Shownet Team spent Sunday through Tuesday configuring, testing, and debugging the network. The exhibition floor opened to conference attendees on Wednesday at noon; by that time, everything was basically working. The two teams spent Wednesday through Friday, while the exhibition floor was open, troubleshooting minor problems that cropped up. After the show closed, pretty much the reverse process took place to disassemble everything.

Lessons

There are a number of lessons to be learned from working on something like the Interop Shownet. First, detailed advance planning pays huge dividends. In particular, if not for the detailed advance planning that went into this project, it wouldn't have been possible to meet the time constraints (especially that the physical network had to be put in place in only 8 hours).

Second, simplicity is important and valuable. For example, as much as possible was done to make the cable plant as generic as possible, by pushing wiring differences all the way to plugs or patch cords at the ends of connections, so that any UTP cable could be used for any purpose, merely by changing what it was plugged in to. Additionally, large and complex tools such as SNMP network management systems were avoided, because they weren't really needed. There wasn't time to establish baselines for SNMP performance monitoring, nor was SNMP mapping of the network required (the network had just been constructed so the Shownet Team and the Core Team knew exactly what it looked like). The Shownet was managed with *ping, telnet, traceroute,* and handheld 2-way radios. That's not appropriate for many situations, but it was here, because the situation was highly dynamic yet very short-lived.

Third, redundancy is also important and valuable. The Shownet designers included lots of extra wire pairs in their plans; they didn't really have a use in mind for these pairs at the time, but they realized that it would be much easier to bundle extra pairs in when the network was being built, and then not use them, than to discover after the network was already hung on the ceiling that they needed more pairs to cope with some unexpected problem or request.

As another example of designed redundancy, there were two UTP drops to each booth on the main exhibit floor, an "Ethernet" drop and a "Token Ring" drop; almost all exhibitors used only the Ethernet drop, and the Token Ring drop was in fact intended all along primarily as a backup Ethernet drop, in case something went wrong with the primary. Finally, all FDDI fiber optic backbone connections were backed up with UTP Ethernet connections, in case the FDDI fiber connections didn't work for some reason.

Fourth, pre-planned fallback strategies are important and valuable. The INTEROP 91 Fall Shownet had several fallback strategies for various services. For instance, if the fiber backbone had failed, a plan was in place to fall back to an Ethernet backbone; if OSPF routing hadn't worked, a plan was in place to fall back to RIP; and so forth.

Conclusion

Working on the INTEROP 91 Fall Shownet was quite a learning experience for me. I had an opportunity to work on and learn about a wide variety of equipment and technologies, as well as to work with and learn from some truly outstanding network planners and managers. Finally, there's a great feeling of accomplishment in seeing the network come together and knowing that you had something to do with its success. If you're interested in being a Shownet volunteer next year, contact Nan Dorio at Interop Company (415-962-2539, `nan@interop.com`).

References

[1] Jacobsen, O., "INTEROP 88 Conference Report," *ConneXions,* Volume 2, No. 11, November 1988.

[2] Almquist, P., "The INTEROP 88 Network—behind the scenes," *ConneXions,* Volume 3, No 2, February 1989.

[3] Dern, D., "Highlights from INTEROP 89," *ConneXions,* Volume 3, No. 11, November 1989.

[4] Knowles, S., "The INTEROP 89 Network: From one of its builders," *ConneXions,* Volume 4, No. 2, February 1990.

[5] Dern, D., "INTEROP 90 Wrap-Up: A View from the Floor," *ConneXions,* Volume 4, No. 11, November 1991.

[6] Knowles, S., "Building the INTEROP 90 Show Network," *ConneXions,* Volume 5, No. 9, September 1991.

[7] Dern, D., "'Dear Cliff': A Report From INTEROP 91 Fall," *ConneXions,* Volume 5, No. 12, December 1991.

[8] Hares, S., "Lessons Learned for OSI at INTEROP 91 Fall," *ConneXions,* Volume 6, No. 3, March 1992.

BRENT CHAPMAN is a consultant in the San Francisco Bay Area, specializing in the configuration, operation, and networking of UNIX systems. During the last several years, he has been an operations manager for a financial services company, a world-renowned corporate research lab, a software engineering company, and a hardware engineering company. He holds a Bachelor of Science degree in Electrical Engineering and Computer Science from the University of California, Berkeley. He can be reached as: `Brent@GreatCircle.COM`.

Status of SNMP Evolution

by James R. Davin, MIT

Framework

The *Simple Network Management Protocol* (SNMP) network management framework is the collection of conventions on which depend the stability, interoperability, and effectiveness of network management mechanisms in the Internet. The framework comprises the specification for the structure of management information (RFC 1155), the definition of a common core management information base (RFC 1213), and the specification of the SNMP protocol itself (RFC 1157). The Internet Activities Board (IAB) conferred Standard status upon these specifications in May 1990, March 1991, and May 1990, respectively.

This standard network management framework has been and continues to be the foundation for a stable, effective network management system in the Internet. Network operators and users daily depend upon the robustness and ubiquity of these deployed mechanisms.

MIBs

The easy extensibility of the framework has been the foundation for enormous growth in the manageability of the Internet. Based on this framework, standardized extensions to the SNMP *Management Information Base* (MIB) have been deployed to manage a variety of network media and devices. Additional standardized extensions to the SNMP MIB (for both new media and new device types) are currently under development within the Internet Engineering Task Force (IETF). The framework has also supported deployment of many private and experimental extensions to the SNMP MIB by which proprietary or advanced functions of the Internet are managed.

Although applications of SNMP technology are increasingly various, the most important function of the SNMP framework in the Internet is to support the monitoring and control of network infrastructure. The design of SNMP is informed by this purpose. Deployment of technology based on the SNMP framework has accrued to network managers and users tangible benefits that derive directly from key architectural principles:

- The SNMP framework minimizes the overall cost of a manageable network by minimizing the cost and complexity of those management system components that are most numerous.
- The SNMP framework fosters ubiquity of deployment by admitting the widest possible range of implementation strategies.
- The SNMP framework fosters operational robustness by realizing management system function as closely as possible to centers of responsible authority.
- The SNMP framework fosters operational robustness by locating control of resources consumed by the management activity (e.g., bandwidth, processing) as closely as possible to centers of responsible authority.

Deficiencies

All human endeavors almost certainly leave room for improvement. Accordingly, informal discussion of perceived deficiencies in the network management framework has arisen from time to time within the IETF community. A special session of the SNMP Directorate at the July 1991 Atlanta meeting of the IETF provided an open forum for such discussion, although the views articulated at that session cannot be said to represent a community consensus or even consistently informed opinion.

A record of that session is available online at IETF repositories as `ietf/snmpdir-minutes-91jul.txt`. (See page 40 for information on how to obtain IETF documents.)

There is little community consensus on what the actual deficiencies of the SNMP framework may be; there is similarly little consensus that any particular change to the framework warrants the attendant operational or architectural impact. At the same time, organization of private efforts towards SNMP evolution have begun among prominent providers of SNMP technology.

Evolution

The most desirable process for evolution of the SNMP framework would be one that combines a prudent regard for the technical and operational aspects of the problem with the fairness and openness required by IETF procedures. Such a process would:

- Provide for community consideration of specific evolutionary directions without biasing the process towards unnecessary or inappropriate change, and;
- Provide for a consensus on all aspects of an evolutionary path that is not dominated by the interests of any particular group.

Written contributions

To begin the process of SNMP evolution, members of the community are hereby invited to submit written contributions that address perceived deficiencies in the SNMP framework. These contributions must be complete and detailed technical specifications that refine, replace, or enlarge upon the current SNMP framework. Each contribution must be such that interoperable implementations could be constructed from it. MIB extensions should not be submitted as SNMP evolution contributions since they are dealt with as non-evolutionary matters in the normal course of events. Contributions to the SNMP evolution discussion should be sent to `internet-drafts@nri.reston.va.us`. All contributions must satisfy the same formatting and copyright conditions applied to all Internet Drafts.

Working Group

When, in the estimation of the NM Area Director and the IESG as a whole, contributions of a sufficient number and quality have been submitted, and when all such contributions have been publicly available for an appropriate deliberative period, a working group will be chartered to consider the submissions. The intent of this two-phase process is to assure that any working group effort will start from the broadest possible range of submissions and that the community will have the opportunity for thoughtful evaluation of all submissions before active working group discussion begins.

By its charter, the working group established to consider proposals for evolution of the SNMP framework will have the option of (a) rejecting any or all contributions as the basis for positive evolution, (b) accepting any or all contributions as candidates for standardization, or (c) modifying or combining any or all contributions to produce consensus proposals for standardization.

The product of the working group will be a single recommendation to the IESG identifying those submitted specifications (or modifications thereof), if any, whose standardization as part of the SNMP framework is agreed to be warranted and desirable. The working group will not be chartered to produce tutorial, explanatory, advisory, or informational documents of any kind. The working group will be disbanded when its consideration of all submitted proposals is complete and its single recommendation is made.

[Ed. Mr. Davin is the Area Director for Network Management in the IETF.]

SLIP Interoperability

by Richard Coop, University of Hull

Abstract

SLIP provides an easy and inexpensive method of connecting TCP/IP hosts with serial lines. It can be utilised to incorporate isolated hosts devoid of LAN technology into a network, provide dialup connectivity where traffic load does not justify use of a leased line and can provide local redundancy in the event of network failure.

Based upon research performed by the author for the UK Internet Consortium, this article discusses several commercial and publicly available implementations of SLIP and dialup SLIP for PCs and *streams* and non-*streams* based SunOS architectures. It highlights implementation and interoperability issues, as well as the importance of Van Jacobson's *header compression* to improve interactive response at low link speeds.

What is SLIP?

SLIP, the *Serial Line Internet Protocol* is a simple data link layer asynchronous framing protocol for transmission of IP datagrams over point-to-point serial and dialup links. As a *de facto* standard and member of the TCP/IP suite of networking protocols, it resides under the IP layer, and being point-to-point requires no network address to physical address resolution and hence has no equivalence of the Ethernet ARP layer.

It was conceived in the early 1980s and was first implemented in 1984 by Rick Adams for 4.2 Berkeley UNIX and Sun workstations [15]. It was quickly seen as an easy, inexpensive and reliable method of connecting TCP/IP hosts and routers with serial lines. Various commercial and publicly available implementations were subsequently developed for Ultrix, SunOS and most other BSD derived systems as well as for IBM compatible PCs.

Why use it?

Almost all UNIX workstations and PCs incorporate ubiquitous serial hardware, mainly RS-232. With the advent of low cost X terminals and workstations capable of supporting increasingly faster serial lines, coupled with the simplicity of the software itself, SLIP can provide a simple and inexpensive communication medium.

Traditionally telecommuters utilised modems and communications software to connect to remote computer systems. Data, however, had to be taken out of packetised form and converted into character form before being transmitted over the dialup telecommunication link. By utilising SLIP, and hence TCP/IP packetised data, multiple sessions can be established to multiple hosts. File transfers are also easier as almost all TCP/IP hosts implement an FTP server.

SLIP can be used to incorporate isolated hosts into a network where they are located in buildings devoid of LAN technology or the cost of utilising such technology is not justified by traffic load. It can be used to interconnect TCP/IP LANs, where gateway machines lack a secondary Ethernet card or the cost of the additional hardware is not justified by the internetwork traffic. In dialup WAN applications, SLIP can provide temporary connectivity when the cost of a permanent dedicated leased line is unjustified.

Deficiencies

The main advantage of SLIP is also its main deficiency; simplicity. The basic SLIP protocol makes no provision for dynamic address assignment, error detection or correction, header compression and only supports the IP protocol family (though others can be supported by encapsulation within IP at the expense of performance). Essentially SLIP provides no way to communicate or negotiate information that may differ between each end of the link.

In order for SLIP to operate, each end of the link must know the other's IP address. Since a SLIP implementation cannot automatically inform the remotely connected implementation of its IP address, static IP addresses must be assigned to both ends. This normally involves two humans verbally agreeing on the addressing scheme to be used. As will be detailed later, header compression can optionally be utilised in some versions of SLIP. Although its specification outlines a methodology for communicating to the other host whether or not it is implementing compressed SLIP, this negotiation procedure must normally be performed manually.

SunOS interoperability tests

Though SLIP is publicly available via anonymous FTP for several UNIX derived systems, due to hardware restrictions the author was limited to implementing SLIP for the SunOS subset. Those investigated and their availability are summarised in Table 3.

Although installation and use of such packages might appear straight forward this was found to be an ill judgement. Some implementations contained erroneous or missing code and coupled with the need to manually configure static SLIP parameters such as IP addresses, installation demanded a reasonable degree of competence on the part of the administrator.

Each SunOS package was installed on the relevant hardware and tested with both itself and each of the others in turn, utilising a "null modem" RS-232 cable between the two workstation ports running at 9600bps. The results of this simple interoperability test are illustrated in Figure 1, where the clear boxes indicate successful interoperability.

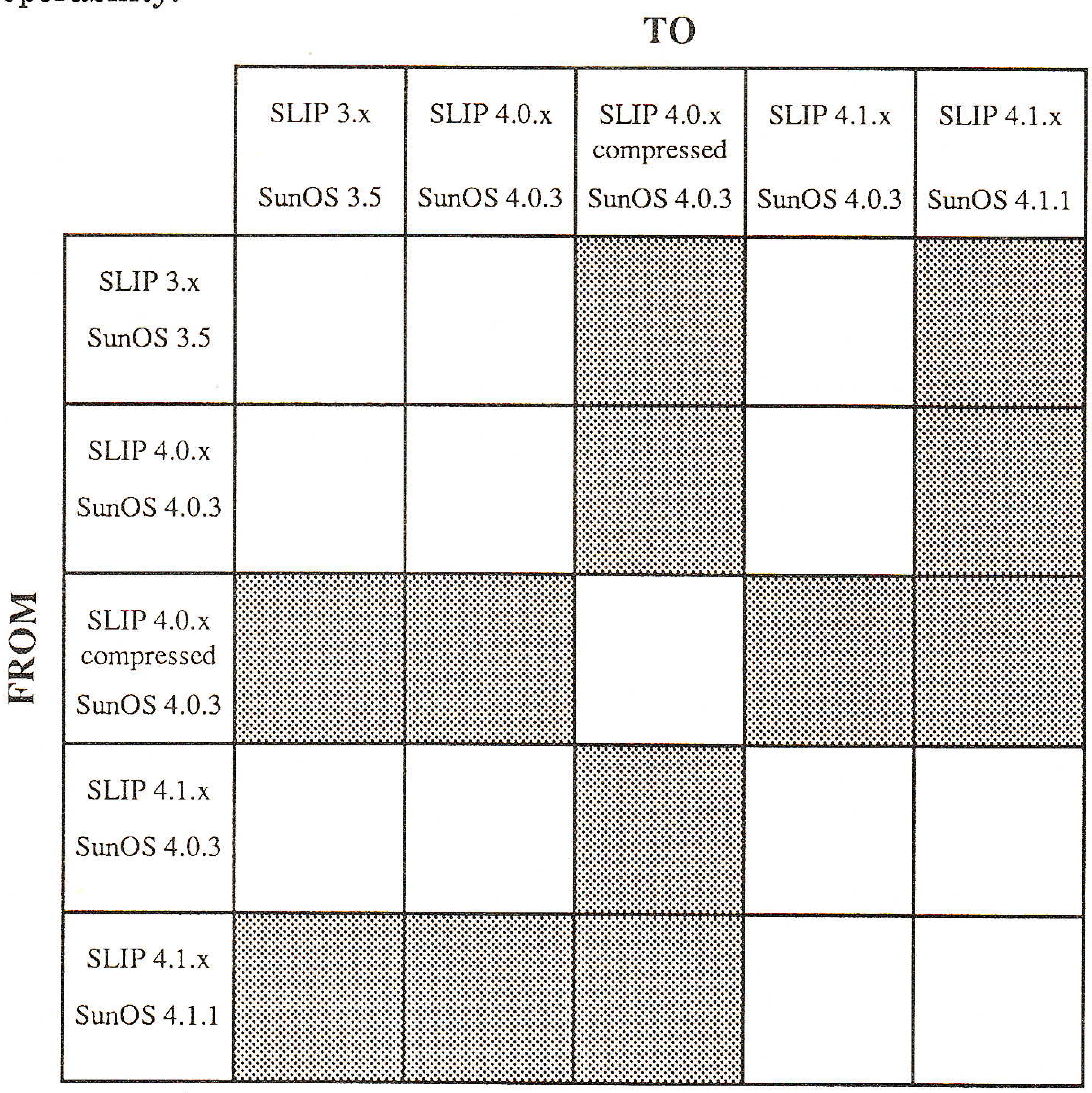


Figure 1: Interoperability of SunOS SLIP implementations

continued on next page

SLIP Interoperability *(continued)*

The shaded boxes indicate unsuccessful interoperability and in these cases the remote host could be *pinged* and the connection established successfully but no *getty* enabled. Although the "connected" message appeared, the connection hung until the FTP or *telnet* application timed out.

The SLIP4.0 implementation is really intended to be used as a login shell for an incoming dialup connection. The *sliplogin* program that comes with the package can, however, be invoked as root to establish the network parameters explicitly and thus establish a local hard-wired link. It was found that the program incorporated an inactivity timer and it was necessary to initially *ping* the remote host once the link had been established, else the route timed out and was deleted.

As SLIP4.0 with header compression is based on SLIP4.0 it exhibits the same problem as well as failing to interoperate with any other package than itself. This latter defect seems to be caused by the implementation failing to implement the full RFC as detailed later.

Ping round trip times were in the order of 210ms. FTP throughput, measured by averaging the throughput of the *ls* command in various directories, was in the order of 0.3Kbytes/s with the exception of compressed SLIP where it increased almost threefold to 0.83Kbytes/s.

PC interoperability tests

There are several versions of SLIP available for IBM compatible PCs. The two utilised were the latest version of Phil Karn's *KA9Q* package and FTP Software's *PC/TCP* generic kernel implementation utilising a *packet driver,* the former publicly available via anonymous FTP for academic and packet radio users, the latter being a commercial implementation [19]. Both were reasonably straight forward to install and their availability is detailed in Table 3.

Both packages were tested for interoperability with a hardwired 9600bps RS-232 link to all the SunOS SLIP implementations previously outlined. In order to communicate successfully, however, it was found necessary to manually reconfigure both KA9Q and PC/TCP to reduce the MSS and window values to 966 bytes (maintaining the *Maximum Transmission Unit* (MTU) value at 1006 bytes) to correspond with those of the SunOS SLIP implementation.

Both KA9Q and PC/TCP were able to communicate with all SunOS SLIP implementations with the exceptions of SLIP4.0 with header compression and SLIP4.1 (Under SunOS 4.1.1), the former being due to SLIP4.0c's inability to send out uncompressed packets. The FTP throughput results for PC to SunOS and vice versa are given in tables 1 and 2 respectively.

FROM \ TO	SLIP 3.x SunOS 3.5	SLIP 4.0.x SunOS 4.0.3	SLIP 4.0.x compressed SunOS 4.0.3	SLIP 4.1.x SunOS 4.0.3	SLIP 4.1.x SunOS 4.1.1
KA9Q	0.39	0.39		0.27	
PC/TCP	0.14	0.22		0.18	

Table 1: PC to SunOS FTP throughput (Kbytes/s)

		FROM				
		SLIP 3.x SunOS 3.5	SLIP 4.0.x SunOS 4.0.3	SLIP 4.0.x compressed SunOS 4.0.3	SLIP 4.1.x SunOS 4.0.3	SLIP 4.1.x SunOS 4.1.1
TO	KA9Q	0.30	0.86		3.70	
	PC/TCP	0.52	0.75		1.10	

Table 2: SunOS to PC FTP throughput (Kbytes/s)

It can be seen than PC/TCP appears to be slower than KA9Q. This can be attributed to the PC/TCP implementation utilising a packet driver, whereas KA9Q communicated with the native serial hardware directly. FTP Software, Inc. supplies a version of PC/TCP for SLIP that works directly with the hardware and if utilised, performance could be considerably higher.

Compressed SLIP

Header compressed SLIP was conceived by Van Jacobson several years ago, motivated primarily by the need to improve typical keyboard echo response during interactive sessions such as *telnet.* Response has been shown to be perceived as "poor" when low level keyboard echo takes longer than 200ms [7]. By reducing the size of the TCP and IP header overhead, the 200ms echo to a character typed can be received over a slower link and the resultant smaller datagrams cause less interference with bulk data traffic.

For most TCP conversations datagram fields for successive datagrams often remain constant or change by a fixed amount and thus redundant header information can be cached by only transmitting the fields which changed from one datagram to the next.

Availability

As far as the author is aware, compressed SLIP is currently available for SunOS 4.0.x as detailed in Table 3 and although available for SunOS 3.x, it is still in the experimental stage of being ported back to a non-*streams* environment.

Though the compressed SLIP for SunOS 4.0.x has options to enable and disable header compression, auto enable and ICMP reject, none of these options seemed to be implemented. The compression code was edited and code added to report the progress of incoming and outgoing packets and whether or not they were compressed. Investigation revealed that although the implementation is prepared to accept normal datagrams it only transmits compressed ones. This is the reason for the interoperability problems experienced in the previous hardwired cases. The two computers exchange normal TCP datagrams to negotiate MSS values and establish the session, but the compressed implementation then only transmits compressed packets causing the other computer to drop these and eventually time out.

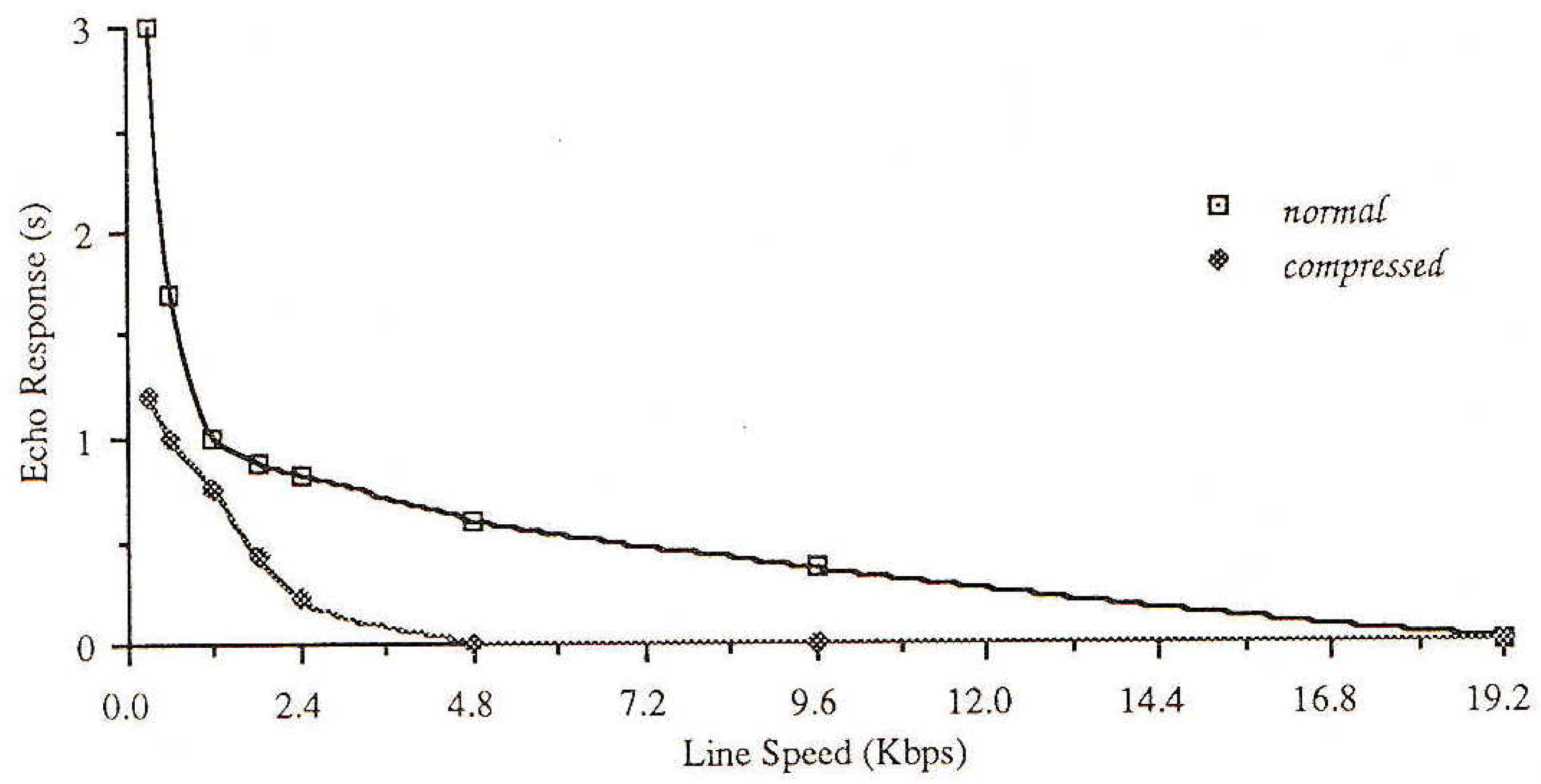
Effect of compression

To investigate the effects of header compression, a Sparc Station 1 running SunOS 4.0.3 was configured with both basic SLIP4.0 and SLIP4.0 with header compression in succession. A SLIP link was established between its two serial ports by the use of a "null modem" RS-232 cable and for each configuration, over a range of available data rates, the average *telnet* echo response and FTP throughput were recorded. As expected, at any given data rate, the *ping* round trip times were found to be approximately equal for both implementations (UDP ICMP packets are not compressible).

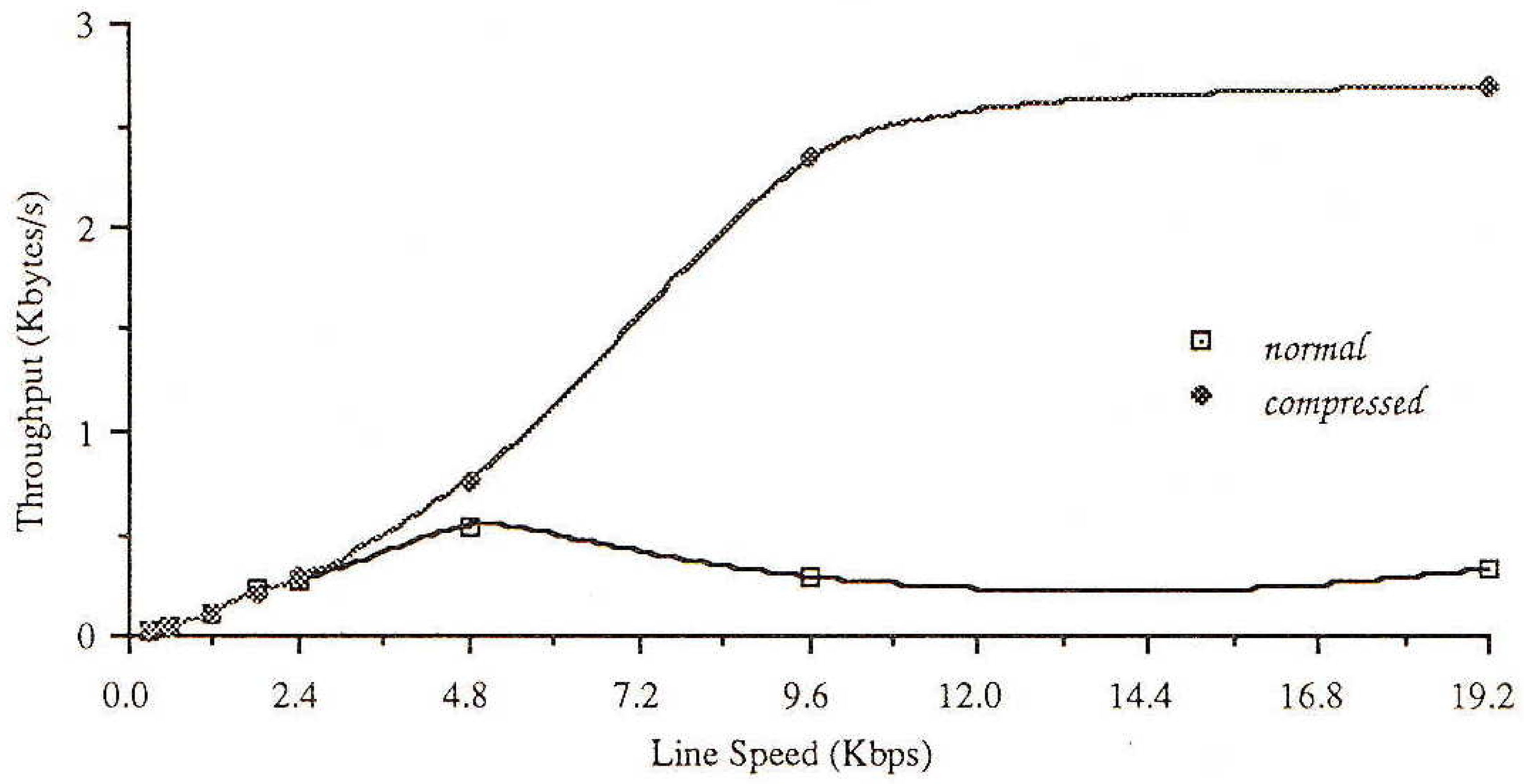
continued on next page

SLIP Interoperability *(continued)*

The results of the effect of header compression on *telnet* echo response and FTP throughput are illustrated by Graphs 1 and 2 respectively.



Graph 1: Effect of compression on *telnet* echo response



Graph 2: Effect of compression on FTP throughput

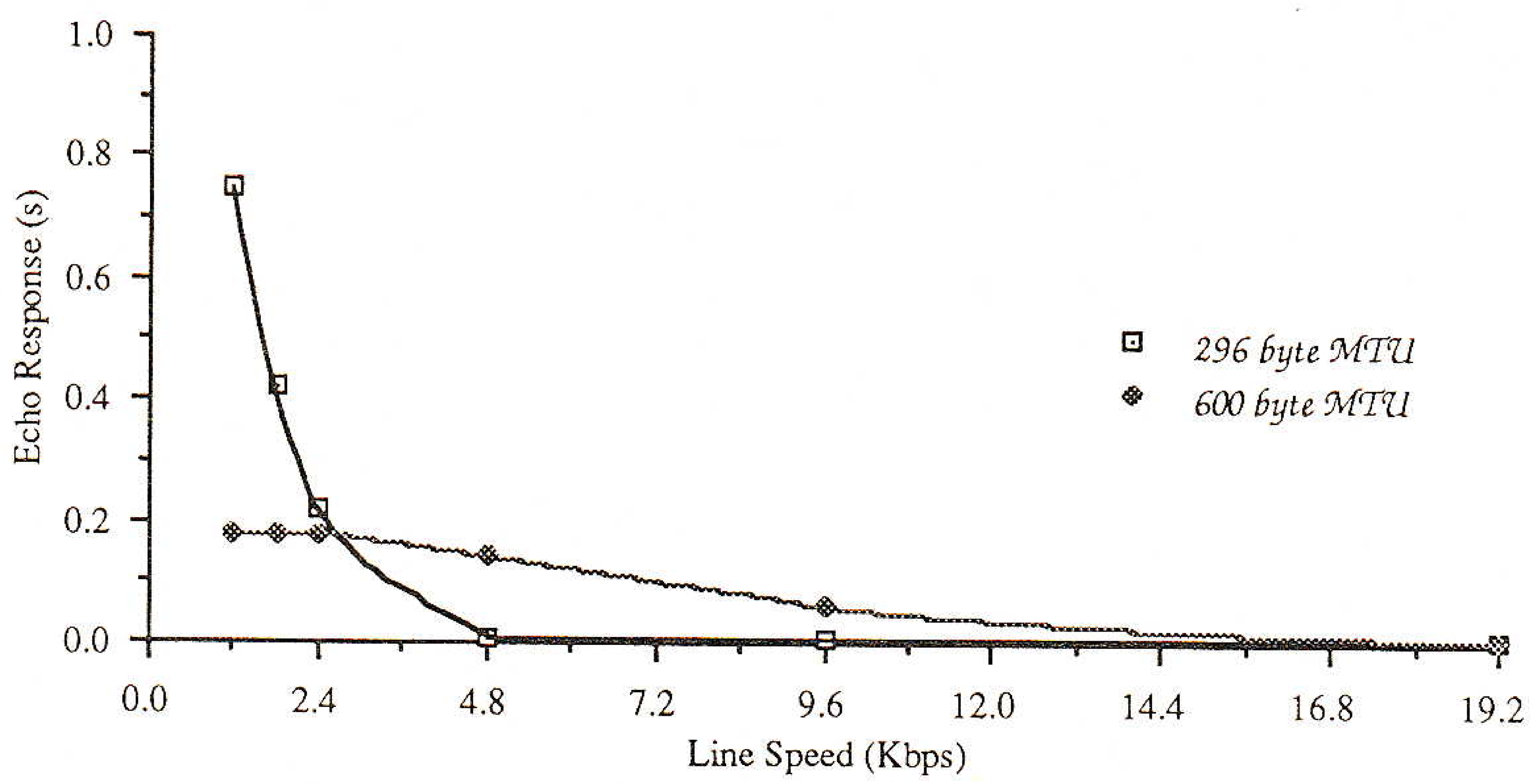
From Graph 1, it can be seen that header compression has a marked effect upon interactive response. Without compression, *telnet* response starts off poor but increases rapidly to a knee point of around 1200bps but then improves slowly, the 200ms acceptance threshold being achieved at a poor 14Kbps. With compression enabled, *telnet* echo response starts off almost 3 times faster and the 200ms threshold is reached at an acceptable 2400bps. Above this point response improves rapidly and at 9600bps response can easily be compared to that of a moderately loaded Ethernet.

From Graph 2, it can be seen that without compression, throughput is linearly related to line speed but then falls off around 4800bps as the *streams* implementation fails to cope with the large datagrams at such high speeds. With the smaller compressed datagrams, however, the *streams* implementation is easily capable of transmission of 19.2 Kbps, reaching a surprisingly fast throughput of around 2.7Kbytes/s.

Effect of increased MTU

As a final test, the effect of additionally increasing the size of the Maximum Transmission Unit (MTU) was investigated for compressed SLIP. To analyse the effect this would have on *telnet* response and FTP throughput, the MTU was increased from 296 to 600 bytes. The resultant effect upon *telnet* echo response is illustrated by Graph 3.

From Graph 3 it can be seen that by increasing the MTU response has become more uniform, the critical point occurring around 2400bps. Below this, *telnet* response is further improved by increasing the MTU size while FTP throughput remains approximately constant. Above this, increased MTU size actually reduced echo response and FTP throughput quite significantly.



Graph 3:Effect of increased MTU on echo response (compressed SLIP)

Dialup SLIP

Dialup SLIP allows TCP/IP to be run over intermittently connected telephone lines and one of the initial limiting factors in its development was that dialup modems in the early 1980s as well as being expensive were too slow to provide acceptable performance. As low cost, high performance 9600 bps modems become available *Dialup IP* became a reality. The author investigated two publicly available packages; *DialupIP* and *tip* with SLIP support.

DialupIP

DialupIP is a publicly available implementation that provides an on-demand dialup IP service based on a modified SLIP driver distributed with 4.3BSD [11]. It provides full IP support over dialup phone lines and runs on Ultrix, SunOS 3.5 and 4.3BSD UNIX platforms and has successfully been used to connect to both *dialupIP* and Toronto's *SLIP4.0* package as well as Phil Karn's *KA9Q* package for PCs [6]. When a datagram is required to be sent, the line is brought up if not up already. Access can be restricted to source host or network, destination host or network, time of day, or protocol. On receiving an incoming dialup connection, security is provided by ensuring the user logs into the system. Only when this has been achieved will control be passed to the *dialupIP* login shell that is used to set up the line discipline.

One disadvantage of such an on-demand *dialupIP* service is that the time taken to dial the modem and set up the line can be very long; longer than the timeout values of applications such as FTP or *telnet.* Dave Smith at `whoops.ftp.com` in the US has found that *dialupIP* fails to deliver DCD dropped signals correctly to the process, and that getting utilities to use the *Domain Name System* (DNS) rather than Sun's *Yellow Pages* is exceptionally difficult without source code. Modem support is also poor; it is only configured for three automatic call units. Work is currently being performed to port *dialupIP* to *streams*-based SunOS systems [6].

Tip with SLIP

Tip with SLIP is a modified version of the *tip* utility running under SunOS 4.0.x that allows the SLIP4.0 and header compressed SLIP4.0 code to be brought up after *tip* has successfully established the connection. It is available by anonymous FTP as detailed in table 3 and provides on-command dialup service that is essentially the partner of SLIP4.0 and header compressed SLIP4.0. *Tip with SLIP* is used to establish an outgoing dialup IP connection and SLIP4.0 provides the login shell for incoming dialup IP connections.

Once again, installation was found to be harder than anticipated. The `login.c` login script processor code required debugging and the *Makefile* required editing to configure the auto call unit in use and to set the path for installation of *man* pages.

continued on next page

SLIP Interoperability *(continued)*

Once installed, it was used to establish an internal 1200bps dialup SLIP connection to another machine running SLIP4.0 using a Miracom HST modem to dial out and a simple Racal MPS1222 modem to answer. However, it was found that *tip with SLIP* locked up the terminal it was invoked on and a route had to be established from another machine connected by Ethernet to the locked workstation in order to communicate across the dialup link. Having established interoperability, the two machines were then rebooted with compressed SLIP4.0. Interoperability was maintained and response was significantly improved as expected.

A 1200bps national dialup link was successfully established between Edinburgh and Hull with the cooperation of Keith Mitchell of Spider Systems, Edinburgh. In Edinburgh a Telebit Netblazer was utilised to invoke the dialup SLIP link and in Hull SLIP4.0 used to provide the login shell for the incoming dialup connection. It was discovered that although the receiving *getty* was asserting parity correctly, the login program did not. It was found necessary to include a "p8" in the *gettytab* entry at Hull to force 8 bit data, no parity on the line. This seemed to resolve the problem and the link operated successfully.

By reconfiguring both ends of the link with compressed SLIP, response was noticeably improved. However, due to possible deficiencies with the Netblazer compression implementation, it was necessary to set its Van Jacobson compression to "on" as setting it to "auto enable" caused it to crash.

Dialup KA9Q and PC/TCP

Both KA9Q and PC/TCP utilise a command to dial a modem and establish a dialup link. It was discovered that although PC/TCP could establish the link, parity was not being asserted correctly and characters could not be interpreted; in the author's opinion the problem lies not with PC/TCP but with the modem utilised. KA9Q was unable even to set up a connection.

PACKAGE	SOURCE(S)	SYSTEM
SLIP 3.x	uunet.uu.net: networking/sl.shar.Z neat.ai.toronto.edu: pub/slipware.tar.Z	4.2 BSD SunOS 3.x
SLIP 4.0	uunet.uu.net: networking/slip/slip-4.0.tar.Z neat.ai.toronto.edu: pub/slip-4.0.tar.Z	SunOS 4.0.x SunOS 4.1
SLIP 4.0 compressed	ftp.ee.lbl.gov: cslipbeta.tar.Z	SunOS 4.0.x SunOS 4.1
SLIP 4.1 beta	dmssyd.syd.dms.csiro.au: pub/slip-4.1.shar	SunOS 4.0.x SunOS 4.1.x
Tip with SLIP	neat.ai.toronto.edu: pub/slipware.tar.Z	SunOS 4.0.x SunOS 4.1
DialupIP	uunet.uu.net: networking/dialup2.0.tar.Z	Ultrix 4.3 BSD SunOS 3.5
KA9Q	thumper.bellcore.com: ka9q/net.exe	PC
PC/TCP	FTP Software, Inc.	PC

Table 3: Sources of SLIP implementations examined

A new standard?

The Internet community realised several years ago that a proposed new standard protocol for serial connections was required. It would address all the deficiencies of SLIP and be flexible enough to address future needs. The result is PPP, the *Point-to-Point Protocol* [13, 17].

However, PPP has taken time to reach its present state of complexity and can prove too complex for most requirements. SLIP, on the other hand, has been around for a long time, is easier to implement and is easier to understand. As a consequence, SLIP became the *de facto* standard for TCP/IP dialup networking.

As TCP/IP implementations become available for increasingly cheaper computers, the use of voice grade dialup network connectivity will further increase. However, as networking becomes more and more complex and the need to transmit protocols other than TCP/IP across asynchronous and synchronous dialup links increases, PPP will replace SLIP as the standard for point-to-point networking.

Unfortunately PPP was not intended to phase out SLIP gradually, and converting from SLIP to PPP can be difficult. It seems contented users of SLIP are presently happy to continue using it until PPP has become established.

Conclusions

It has been shown that publicly available implementations of SLIP exist that can be utilised to provide simple and inexpensive hardwired and dialup IP connectivity, and that header compression can significantly improve the performance of the basic protocol.

Although the concept of SLIP is fairly simple, it should be stressed that implementation was found to be more difficult than first anticipated. Coding was found to be erroneous and much of the documentation assumed a fairly high degree of UNIX competence on the part of the administrator. In addition, network administrators must be prepared to manually negotiate and agree on parameters such as local and remote IP addresses and the link data rate.

As TCP/IP implementations become available for increasingly cheaper computers, the use of voice grade dialup network connectivity will further increase. However, as networking becomes more and more complex and the need to transmit protocols other than TCP/IP across asynchronous and synchronous dialup links increases, the Point to Point Protocol will replace SLIP as the standard for point-to-point networking.

Acknowledgements

I wish to thank my M.Sc. supervisor, Brian Tompsett, for stimulating ideas leading to a clear and succinct research conclusion, for suggestions of reading material and cited mailing lists and for encouragement and support in producing papers. Acknowledgement is due to the UK Internet Consortium for help at all stages of the research. I wish to thank in particular Dwight Ernest, Hyphen Editorial Systems for all his support, Caroline Bowden, Unipalm Ltd. for the loan of PC/TCP for evaluation and Keith Mitchell, Spider Systems for the loan of a Miracom modem and cooperation in establishing a dialup SLIP link. I wish to further acknowledge Ole J. Jacobsen, editor and publisher of *ConneXions—The Interoperability Report,* and the editor of *UNIX Today!* for kindly mailing relevant material as well as Alex Sharaz, Dave Renardson, Paul Foster, Tony Foster and Mark Bell at the University of Hull for the loan of a Racal modem and numerous serial cables.

continued on next page

SLIP Interoperability *(continued)*

References

[1] R. Coop, "Investigation of SLIP for Hardwired and Dialup Links," M.Sc. dissertation, Department of Computer Science, University of Hull, November 1991.

[2] R. Coop, "Investigation of Current Implementations of SLIP for SunOS and Personal Computers," Proc. UK UNIX Users Group Conference, Heriot-Watt University, November 1991.

[3] R. Coop, "SLIP/PPP Dialup," *IPNETWORKING: The Journal of the UKIC Users Group,* Volume 1, No 1, 1992.

[4] R. Coop & B. Tompsett, "An Investigation of SLIP and Dialup SLIP," Research Report 91/3, Department of Computer Science, University of Hull, November 1991.

[5] R. Coop & B. Tompsett, "Current Commercial and Public Implementations of SLIP, Dialup SLIP and PPP for SunOS UNIX Systems and IBM Compatible PCs," Report for the UK Internet Consortium, Department of Computer Science, University of Hull, November 1991.

[6] `Dialupip-request@sh.cs.net` mailing list.

[7] V. Jacobson, "Compressing TCP/IP Headers for Low Speed Serial Links," RFC 1144, February 1990.

[8] P. Karn, "NET User Reference Manual (NOS Version)," 15/3/91.

[9] B. Garbee, P. Karn and B. Lloyd, "The KA9Q Internet Software Package—Preliminary Version," 20/1/88, © 1988 B. Garbee, P. Karn and B. Lloyd.

[10] J. Levitt, "Two Protocols Worth Observing," *UNIX Today!* 7/1/91.

[11] C. Partridge, "Dialup IP," *ConneXions,* Volume 3, No 11, November 1989.

[12] "PC/TCP Network Software for DOS," User Manual, 1990, FTP Software Inc., 26 Princess Street, Wakefield, MA 01880.

[13] D. Perkins, "The Point to Point Protocol: A proposal for Multi-protocol Transmission of Datagrams over Point to Point Lines," RFC 1134, 1989.

[14] D. Perkins & R. Hobby, "The Point to Point Protocol Initial Configuration Options," RFC 1172, July 1990.

[15] J. Romkey, "A Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP," RFC 1005, June 1988.

[16] `Sun-managers-request@nsfnet-relay.ac.uk` mailing list.

[17] R. Hobby, "The Point to Point Protocol (PPP)—A new proposed standard Serial Line Protocol," *ConneXions,* Volume 4, No. 4, April 1990.

[18] J. Romkey, "SLIP: Serial Line IP," *ConneXions,* Volume 2, No. 5, May 1988.

[19] J. Romkey, "The Packet Driver," *ConneXions,* Volume 4, No. 7, July 1990.

RICHARD COOP gained his first degree in Electrical and Electronic Engineering at the University of Hull (B.Eng. 1990). He developed an interest in Computer Science, in particular networking, and after graduating went on to pursue an M.Sc. at the University of Hull. For his dissertation he chose to perform a detailed study for the UK Internet Consortium into practical implementations of SLIP for dialup connectivity.

Opinion: OSI Is (Still) a Good Idea

by Richard desJardins, The GOSIP Institute™

Introduction

OSI is the *de jure* international standardization component of worldwide computer networking. It is a joint program of ISO and CCITT, as well as the major world governments. OSI is used in public systems such as the world telecommunication networks, the public message handling system, and public directory services, as well as the US and UK *Government OSI Profiles* (GOSIPs) and the *European Procurement Handbook for Open Systems* (EPHOS).

The Internet is the worldwide public domain internetwork. The Internet runs not just TCP/IP, but is multiprotocol and inclusive. TCP/IP is accepted worldwide as the public domain (*de facto*) standard for computer network interoperability. The Internet has done a lot of things right.

OSI has done a lot of things right too. Everyone uses the Reference Model to compare protocol architectures. Many architectures use the OSI *Service Definitions,* or at least the concept of service definitions. Neither of these two things were ever successfully developed in TCP/IP because the Internet was out implementing and deploying protocols, and the architecture and service definitions were just, well, whatever they were. (Who cared? The entire community was a few researchers who worked together and learned as they went.) Now the Internet is much more formal, and has its own standardization procedures, which are becoming more ISO-like (no surprise).

The Internet recognizes OSI standards today (e.g., IEEE 802 LANs, CLNP, Object Identifiers, X.500, all of which are International Standards), along with proprietary standards such as Novell IPX and AppleTalk, along with consortia standards such as The X Window System and Unix International, along with its own RFC standards. The Internet protocol suite will continue to recognize OSI standards.

The Internet Society has become the public domain network standardization body, and its RFCs are recognized as public domain standards in their own right. This acceptance has accelerated the Internet trend toward multiprotocol accommodation and inclusivity. The *Internet Activities Board* (IAB) now has the same responsibilities and the same due process requirements in their approval of standards as the ISO Council or the CCITT Plenary.

Bigots

This emergence of TCP/IP as a peer to ISO and CCITT is all to the good, just like the convergence of cultures and races in the world at large: the only people who object to it are the bigots, whether they are spelled "OSI Bigots" or "IP Bigots." (As someone said at the recent ROAD meeting, "Unanimity might be achieved—if we shoot a few people.")

I say, "let's make a deal." If we (OSI) admit that some of our stuff (Session Layer, for one, which Marshall Rose calls "The Sewer of OSI") is not so good, will you (IP) admit that some of our stuff is really the best solution and should be used in the Internet, even if it's spelled 'OSI'? Then let's continue to get the people of good will from both communities to work together to find the best solutions, whether they are two-letter words or three-letter words, and let's just line up the bigots against a wall and shoot them.

(desJardins' First Law of Protocol Naming: Ya can call it "IP" or ya can call it "TCP/IP," but ya doesn't has ta call it "Internet.")

continued on next page

OSI Is (Still) a Good Idea *(continued)*

Specific Arguments and 7 Fearless Predictions

Seven is not a magic number, but it works for me:

1. The IP Bigots are spinning like dervishes to make IP solve all the future address growth and routing table growth problems, simply because they hate to use anything spelled "OSI." But the true fact is, CLNP can readily solve both those problems, because CLNP is just IP with a big address space. With CLNP, systems can have two or three addresses, e.g., an IP address (if they get one soon, before they run out), a network address, and a GOSIP organization "address" (which is really an organization global name plus an internal address, but is not a global address). The inter-domain routers (after all, there aren't really so many of these) can be smart about how to handle the different addresses to cut down on routing table size, implement policy routing and type of service routing, etc.

CLNP

Fearless Prediction #1: CLNP will be accepted as "IP–2." (We may have to shoot some people, but, hey, the Indians got in the way of Manifest Destiny.)

Routing

2. As long as we're talking CLNP, let's face it, OSI routing protocols—i.e., ES–IS, IS–IS, and IDRP—are as good as or better than IP routing protocols, because OSI was able to apply all the lessons learned from years of making IP routing protocols work. So the IP Bigots have to listen to this: CLNP, ES–IS, IS–IS, and IDRP are really good stuff!

Fearless Prediction #2: IDRP will be used as the single inter-domain routing protocol for both IP and CLNP.

CONS

3. Roll over, Beethoven, because *Connection Oriented Network Service* (CONS) is not dead yet. (CONS is not really spelled "X.25," folks.)

Fearless Prediction #3: The network paradigm is going to shift again by the end of the decade: Transport service will use CONS when it makes economic sense for point-to-point and isochronous traffic, and CLNS when it makes sense for multi-point and bursty traffic. (And speaking of Transport, there will be a "TP5" some day soon, with screaming performance—would you believe 1Gbps?—based on rate control and selective retransmission on top of CONS environments.)

Implementors

4. A word about "implementors": A lot of the trouble with OSI is with the implementations, not with the standards: In too many cases, they're expensive, not terribly robust, and poorly integrated with the rest of the product line. For example, why don't we see "skinny" upper layer stacks, which OSI readily allows? Because in many cases the implementors built the upper layers as three protocol machines, each with full functionality, no matter whether applications needed it or not! OSI wasn't *designed* like that, it was *implemented* like that. And the biggest implementation of them all is ISODE, which is a performance and memory hog. (This is appropriate for its mission, which is a development environment, not intended as a production system, but some of the "implementors" from the vendors just took it and productized it.) Not the way to do it, guys! Talk to Jim Quigley (chair of the NIST OIW Upper Layers SIG) about the OSI *Skinny Stack*, which arose from work in ANSI and EWOS to run the X Window System over OSI [1]. As implemented at the University of London Computing Centre, the Skinny Stack can run many of the current and future OSI applications with 2K lines of code (compared with the 30K+ lines of code in many if not most OSI upper layer implementations). TCP/IP code comes free with UNIX, whereas you have to pay for OSI. Which one would you buy?

Fearless Prediction #4: OSI will come free with UNIX by the end of the decade, and will finally be accepted when Distributed Transaction Processing and Remote Procedure Calls run like banshees over the OSI Skinny Stack.

(Let's put out an RFC for the Skinny Stack, and fast-track it through ISO. It takes only months, not years. That should make Marshall Rose happy, and it is certainly a needed addition to ISODE! Let's make the skinny profile a NIST OIW, ISO ISP, and GOSIP standard. Aside to the OSI Regional Workshops: Why didn't we do this three years ago?)

Cheap standards

5. Let's get the standards published cheap, or free. Standards should be the price of a technical book, or cheaper, and it should be legal to make copies of them and FTP or FTAM them across a network. According to some copyright experts, ISO and CCITT don't "own" their standards, they just claim they do. Their standards are developed in the public domain, then just before being published are slapped with a copyright notice so that you have to pay their prices. (By the way, FTPing standards across the Internet is not "free," you just don't have to pay for it. It costs about the same as printing and mailing a paper copy.) So ISO, CCITT, and Internet standards should all be freely available at low or no cost. The way to make this happen is happening now: The Internet Society is effectively challenging the established standards bodies to open up their process. I think it's going to work.

Fearless Prediction #5: By the end of the decade, the Internet, ISO, and CCITT are going to be working together to publish the standards for the Worldwide Internet at low or no cost. (As Carl Malamud might say, still smarting from his aborted experiment to make the CCITT Recommendations available on the Internet via Anonymous FTP, "Right, and monkeys are going to fly out of my butt!")

The Future

6. Fearless Prediction #6: The future of network computing is: Applications will run on top of APIs, which will plug into XTI/TLIs, which will run over various Transports, which will run over CONS and CLNS, which will all come free in your UNIX boxes. (We're talking about a few hundred lines of code here and there, come on, guys, don't make people pay extra money for this!) X/Open will continue to occupy a key role here. By the end of the decade, UNIX software will be distributed on 10 CD ROMs, of which half will be for The X Window System and MOTIF. (Just kidding on that last one, I think.)

The test

7. The test of a good standard will become (as it should be): Does it provide the needed functionality? Is it cheap to implement? Does it work as a good citizen in the Internet? Not: Is it spelled "IP"? Or: Is it spelled "OSI"? And speaking of testing (was I?):

Fearless Prediction #7: "One world testing" (i.e., test a product just once to certify it for the worldwide ISO/CCITT/Internet market) will be done with a modestly sized suite of worldwide standard tests for each protocol standard. Maybe COS, SPAG, and INTAP can get together and figure out how to do this at a reasonable cost. (Please?)

[1] Peter Furniss, "The OSI Skinny Stack," University of London Computing Centre, March 1992.

RICHARD desJARDINS is education director of The GOSIP Institute. He is two-term immediate past chairman of ISO Subcommittee 21 on OSI and he was one of the original key contributors to the OSI Reference Model. He has served as a senior systems engineer with NASA, DARPA, and CTA Incorporated for over 25 years. He has an M.S. in computer science from the University of Maryland as well as degrees in mathematics and physics from Catholic University.

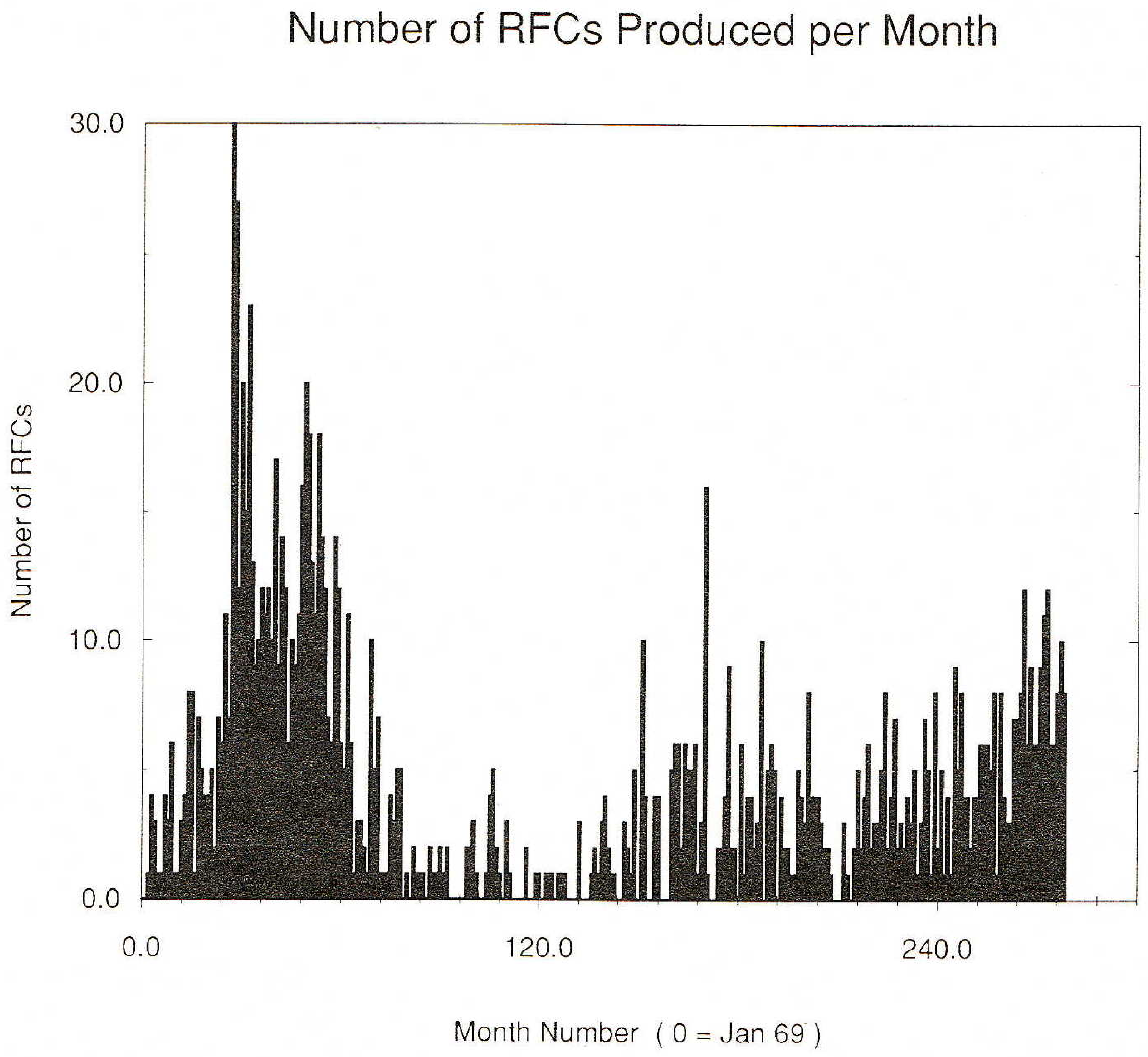
Letters to the Editor

Ole:

In looking at the nicely humorous article by Jon Crowcroft in the May 1992 (Vol 6, No 5) issue of *ConneXions,* I found that the chart seemed to me to be a bit at odds with my subjective memory of the variations in the production of RFCs. I certainly did not recall that the production rate was so high in the early 70's, and i was sure the rate was quite a bit higher than shown in the chart in the last few years.

I decided to draw a graph based on data from the "`rfc-index.txt`" file provided by the SRI International Network Information Service Center (SRI-NISC). This is in close agreement with Jon Crowcroft's chart for about the first 2/3's of the data, but is substantially different in the last 1/3.

—Jon Postel, The RFC Editor, USC-ISI



Ole,
I concur with Jon Postel's correxion—on looking at the raw data, my *awk* script must have gone awry.

—Jon Crowcroft, UCL

We appreciate the feedback and the corrected histogram. It just goes to show what you can do with some statistics and a graph. Perhaps someone will send us the equivalent histogram for OSI documents.

—Ed.

My opinion article "The Trouble with OSI," in the May issue of ConneXions triggered quite a few reactions. In addition to the article on page 33 of this month's issue (and promises of more articles from other quarters), we received some letters (all of them from "Bob" for some reason :–)

Dear Ole,
Just read your opinion on the trouble with OSI, and I agree. I see TCP/IP and OSI merging, with bigots on both sides mellowing out to the good of all. For example, I see X.400/X.500 coming in over TCP/IP. Now, where does ISDN fit in? And, while you're up, how about HDTV?

—Bob Metcalfe, InfoWorld

Ole,
I don't basically disagree, but I would like to see an important point made about the failure of OSI:

The tradition of the Internet is that standardization *follows* implementation. Also the people putting the effort into doing experimental implementations of draft standards get a lot of say (including a veto on extra features) as the standardization process comes down to the wire.

My recent experience of the IETF standardization process suggests that there is a danger from people drifting over to the IETF process from the OSI standardization world who do not support this tradition. Since I think that tradition is fundamental to the success of the Internet, I would like to see it emphasized at every possible opportunity.

—Bob Smart, CSIRO, Australia

Ole:
I found much to agree with in your "Opinion: The trouble with OSI" op-ed piece in the May *ConneXions*. OSI did indeed bite off a very large mouthful, probably more than we all realized at the time [I was the editor of X.200 during the 1980–84 Study Period]. Generally, there was always some regional standards group that arrived at every meeting with a partially developed protocol which they would push very hard. Of course the NIH factor would come into play and there would be another regional group with YAP (*Yet Another Protocol*).

There was not only the mistrust (net bigotry?) between the ISO camp and the Internet people, but also the mistrust and mythological differences between the computer and telecom people. Also, there are many people who are unaware of how the standardization process works. In the United States, under ANSI rules, anyone may participate and those with the knowledge should participate. Instead, those of us who went to all of those exotic cities, to the envy of our fellow workers, got to do the work. Regarding most of those cities, I can show you the path from the hotel to the meeting place and little else as there was no time for sightseeing.

As to standards being under constant revision, in 99% of the cases all those revisions must be backward compatible. In CCITT we had to assure this since the carriers and administrations had to grandfather all existing customers. Sadly enough, neither ISO nor the CCITT have ever "gone online" with this work in an effort to expedite the work. Standards making is part technology, part marketing, and part diplomacy—not a simple job description. In my 10+ years involved with the process, I met very many people for whom I have a great deal of respect. The long hours, the travel, and the preparation are demanding. I keep thinking that one day I may write a book on how many marriages have broken due to the enforced separation. I am not suggesting that you "Take a standards person to lunch," but all people involved in this area should make themselves aware of the process and find out if and where they can contribute. There certainly is no shortage of work, just a shortage of workers.

—Bob Blackshaw, Corporation for Open Systems International

Book Review

Open System LANs and their global interconnection, by J. Houldsworth, M. Taylor, K. Caves, A. Flatman, and K. Crook, Butterworth Heinemann, 1991, ISBN 0-7506-0145-X (Paperback).

Serendipity. I was in the process of compiling ISODE 7.0 and contemplating installing PP, when I came across this book. I didn't find it particularly useful in getting ISODE up and running, but it was an interesting book nevertheless.

Organisation

It seems that most books about OSI are meant for reference, and at first glance, this looked to be no exception. The book starts off nice and easily with a brief introduction to LANs and WANs in the first chapter and a basic introduction to the seven layer model (though, surprisingly perhaps, very little explanation of the reasoning behind the seven layer model).

The book then goes on to describe the standards in detail starting at the physical layer and working up. (Why can't people be more original?) The chapter on LAN Standards covers Ethernet, Token Ring, Slotted Ring, FDDI and MAC bridging, all fairly comprehensively—most sections including cabling specification, signaling topology considerations and management of each particular type of network. The coverage of the data link layer, network layer (which includes X.25) and transport layers are equally thorough, with chapters devoted to each.

Comparison

The authors then attempt to briefly compare OSI to some other networking standards, namely TCP/IP, Microsoft's *LAN Manager* and Novell's *NetWare.* After that come chapters on Network and OSI Management, and structured building cabling. The book ends with a chapter on future directions of OSI, which includes information on B-ISDN, ATM and FDDI-II.

Evangelism

The main problem with this text is the occasional habit of the authors of lapsing into evangelising. This is particularly marked when they are talking about OSI in relation to other protocols. However, where the book avoids this, it is clearly written and comprehensive, with a high signal to noise ratio. Where the authors avoid excessive use of acronyms (hard in an OSI text), its readable too. Even when you get stuck, you've got 2 appendices (28 pages) of glossary and referred-to standards.

Cabling

The comparison chapter is a little out of place—the comparisons are only very basic and don't do justice to the non-OSI protocols, especially TCP/IP. The book would probably be better without it, and it should have just concentrated on defining OSI. I was pleasantly surprised to find the chapter on cabling in your building, because you don't usually find useful information like that in this sort of book.

Will I keep this book on my shelf? Yes, I think so. Will I use it? Again, probably.

—Matthew Farwell, UK IBM PC User Group

The NEARnet Trouble Ticket System

Bolt Beranek and Newman is pleased to announce the availability of The *NEARnet Trouble Ticket System*. The system allows problems and their related working notes to be maintained in a coordinated fashion, and provides for automatic distribution of ticket information via electronic mail.

System overview

The system is built on the INFORMIX Relational Database running on a Sun Sparcstation. It uses the Embedded-SQL (in **C**) package to interface to the mail system (currently MMDF). The system includes the necessary definitions for the INFORMIX "forms" front-end, which is used for ticket data entry and searching. The system also includes **C** code to provide finger-based access to tickets in the database.

Design

The system grew out of a weekend hack (as most useful things do). It has evolved as people have requested new features. As such, it suffers from a lack of "clean design" and has a fair amount of NEARnet/ MMDF-specific stuff in it. There are several things I would have done very differently if I'd known other folks would be looking at the guts of it :–).

In practice, however, this system has been immensely useful to us and has helped NEARnet gain a reputation for, among other things :–), persistent and thorough problem resolution. I hope that this system, while it may not be immediately useful in your environment, will at least serve as a model for the kind of thing that can be built around an off-the-shelf database package.

Current release

The current release package was prepared by Leo Dopson and John Curran. It contains several descriptive documents in the `docs` directory and an easy-to-use installation script that will customize the system to your network's requirements: `bin/install_ticket_system`.

In the future, we hope to provide improvements to the system, including making the package more tailorable, adding *Sendmail* support, and adding more of the features proposed by the IETF *User Connectivity Problems Working Group*. To facilitate this process, please feed any improvements you make to this package back so we can all benefit.

Availability

The system is available via anonymous FTP on `nic.near.net` in the file:

`pub/nearnet-ticket-system-v1.2.tar`

Mailing list

Bug reports, discussion, fixes, improvements, questions should be addressed to:

`tt@nic.near.net`

To join this list, mail to:

`tt-request@nic.near.net`

—Dan Long
Senior Network Analyst
BBN Systems and Technologies

Getting Internet-Drafts

Internet-Drafts

Internet-Drafts documents represent work-in-progress of the Internet Engineering Task Force (IETF). It should be noted that Internet-Drafts are *not* standards and systems should never be cited as "in compliance with" the drafts. Internet-Drafts are available by anonymous FTP. Login with the username "anonymous" and password "guest." After logging in, type "cd internet-drafts" and "get <filename>." For example:

"`get draft-ietf-osids-simple-stack-00.txt`" or

"`get draft-ietf-osids-simple-stack-00.ps`"

Internet-Drafts directories are located at:

- *East Coast (US):*

  Address: `nnsc.nsf.net (128.89.1.178)`
- *West Coast (US):*

  Address: `ftp.nisc.sri.com (192.33.33.22)`
- *Pacific Rim:*

  Address: `munnari.oz.au (128.250.1.21)`
- *Europe:*

  Address: `nic.nordu.net (192.36.148.17)`

Internet-Drafts are also available by e-mail. Send a message to: `mail-server@nisc.sri.com`. In the body type:

"`SEND internet-drafts/draft-ietf-osids-simple-stack-00.txt`" or

"`SEND internet-drafts/draft-ietf-osids-simple-stack-00.ps`"

For questions, please mail to: `internet-drafts@nri.reston.va.us`.

Getting Request for Comments (RFCs)

RFCs

Having told you how to obtain Internet-Drafts, we should quickly add a note about *Request for Comments* (RFCs), the official Internet document series. RFCs can be obtained via anonymous FTP from a number of repositories including:

- `NIC.DDN.MIL`
- `FTP.NISC.SRI.COM`
- `NIS.NSF.NET`
- `NISC.JVNC.NET`
- `VENERA.ISI.EDU`
- `WUARCHIVE.WUSTL.EDU`
- `INFO@SH.CS.NET`
- `sunic.sunet.se`
- `walhalla.informatik.uni-dortmund.de`
- `mcsun.eu.net`
- `funet.fi`
- `ugle.unit.no`
- `ftp.diku.dk`

More information

Some of these sites provide an automatic "RFC by e-mail" service. Hardcopy versions of the RFCs can be purchased from the Network Information Systems Center at SRI International, send e-mail to `nisc@nisc.sri.com` or call 1-415-859-6387. For more details about how to obtain RFCs, see *ConneXions,* Volume 6, No. 1, January 1992.

Preliminary Announcement and Call for Papers

The *International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (MASCOTS '93) will be held January 17–20, 1993 at the Hyatt Hotel, La Jolla, San Diego, California, USA. The workshop is part of the 1993 SCS Western Multiconference on Computer Simulation and is sponsored by the SCS, IEEE TCSIM, and IEEE TCARCH. (ACM, IEEECS, IFIPWG, ORSA cooperation/sponsorships have been requested).

Scope

The workshop is expected to be a major event where researchers, developers and experts with interests in systems design, modeling and analysis, simulation, performance evaluation and various applications will meet to consider one of the current important themes: modeling, analysis and simulation of computer/communication systems of the present and future.

Performance, robustness and reliability predictions of computer and communication systems of future in particular are both important and extremely challenging. Modeling, analysis and simulation are widely applicable to problems in the specification and design of computer and communication systems; however, traditional modeling, analysis and simulation strategies need to be closely scrutinised for their robustness, efficiency and practical applicability in areas of future computer/communication systems. Some techniques are evolving and new approaches are emerging to satisfy the requirements of these ever-expanding areas.

Topics

Topics of interest include, but are not limited to:

Modeling / Analysis / Simulation of Systems such as:

- Multiple Processor Systems
- High-Speed Computer Networks and Distributed Systems
- Massively Parallel and Scalable Systems
- Systolic structures and SIMD/Vector Machines
- Fault Tolerant Systems
- Real-Time Systems
- Artificial Neural Networks
- Parallel, VLSI and RISC/CISC architectures
- Large, Distributed, (Incomplete) Data-base Systems
- Application systems like DSP/AI applications and expert systems/Vision and Image Processing Systems/Robotics and Control etc.
- Complex and heterogeneous systems
- Novel architectures/advances in technologies
- Telecommunication/communication systems

Advances in Modeling Techniques such as:

- Analytic (Performance, Reliability and Performability) Modeling
- Specification and validation techniques as in network protocols, logic design etc.
- Discrete Simulation
- Numeric Simulation and Visualization Techniques
- Intelligent Simulation Techniques with Knowledge as the key

continued on next page

Announcement and Call for Papers *(continued)*

Author information

Papers that deal with these themes, both methodological and specific case study-oriented, are solicited. We are especially interested in papers on innovative modeling and/or simulation techniques that are expected to survive the onslaught of technological advances and remain current for a reasonable period of time.

The workshop will have prominent guest speakers, presentations of refereed papers, panel sessions, tool and poster presentations. In addition, there will be tutorials on introductory and advanced topics.

All submissions will be reviewed. We shall provide blind referring. Put names, affiliations and addresses for correspondence (postal and electronic) of authors on a separate cover. Papers must not exceed 12 double-spaced pages. Provide an abstract and 3-4 keywords. Authors of accepted papers will have to present their papers at the workshop. A best paper award will be made.

A limited number of posters/short papers can also be admitted. For posters, an extended abstract of maximum 4 double-spaced pages must be submitted. Full-length papers and poster abstracts will appear in the proceedings to be published by the SCS.

Proposals for tutorials must contain a brief description of the theme, target audience, proposed time duration and an outline of the presentation. Panel proposals should be based on a current theme that can generate a lively discussion. A brief position statement from the each panelist of a panel will be included in the proceedings.

A selected set of high-quality papers, tutorials, state-of-the-art and invited lectures (revised and enlarged) will be published as a hard cover book, by Kluwer Academic Publishers, shortly after the workshop. Send four copies of your papers and posters to the Program Committee Chair, Jean Walrand at the address given below. Panel session proposals and proposals for tutorial, should be sent to the respective Chairs. Special session proposals should be directed to Kallol Bagchi, Technical Co-Chair. All other inquiries should be directed to the Publicity Chair.

There will be a Tools Fair at *MASCOTS'93*. Tools accepted for presentation should be demonstrated on a computer system or shown on a video tape. All submissions on tools must contain the details of the necessary equipment. Send abstracts on tools to one of the Tools Fair Managers. Brief descriptions of selected tools including pictures (max. 1 double-spaced page) will also appear in the proceedings.

Important dates

June 30, 92:	Special sessions proposals due
July 15, 92:	Deadline for submissions of papers/tutorial and panel proposals/tools
Sept. 15, 92:	Notification of acceptance
Oct. 15, 92:	Camera-ready copy due

Conference Chairs

General Chair:

Dr. Herb Schwetman
MCC
3500 West Balcones Center Drive
Austin, TX 78759, USA
Phone: +1 512-338-3428
Fax. +1 512-338-3885
E-mail: hds@mcc.com

Program Committee Chair:

Prof. Jean Walrand
University of California
267M Cory Hall
Berkeley, CA 94720, USA
+1 510-642-1529
+1 510-643-8426
wlr@diva.berkeley.edu

Technical Co-Chairs:

Dr. Kallol Bagchi
Aalborg University,
Fredrick Bajers Vej 7
DK-9220, Aalborg Øst
DENMARK
Phone. +45 98 15 85 22
E-mail: kkb@iesd.auc.dk

Dr. Doug DeGroot, MS 8435
Texas Instruments
6550 Chase Oaks Blvd.
Plano, TX 75023
USA
+1 214 575 3763
degroot@dog.dseg.ti.com

Tutorial Chair:

Prof. Kishor Trivedi
Duke University
Durham NC 27706
USA
E-mail: kst@egr.duke.edu

Panel Chair:

Prof. Dharma Agrawal, Box 7911
North Carolina State University
Raleigh, NC 27695-7911
USA
dpa@csl36h.csl.ncsu.edu

Publicity Chair:

Dr. Patrick Dowd
State University of New York
Buffalo, NY 14260
USA
E-mail: dowd@eng.buffalo.edu

Treasurer:

Dr. Anup Kumar
University of Louisville
Louisville, KY 40292
USA
AOKUMA01@ulkyvx.bitnet

Tools Fair Managers:

Dr. Thomas Braunl
IPVR
University of Stuttgart,
Breitwiesenstr. 20–22
D-7000 Stuttgart 80,
GERMANY
E-mail:
braunl@informatik.uni-stuttgart.de

Dr. Manu Thapar
HP Research Labs.
Hewlett Packard Company
1501 Page Mill Road
Palo Alto, CA 94304
USA
thapar@hpl.hplabs.hp.com

More information

If you are interested in receiving further announcements of this workshop, please contact (preferably by e-mail):

Dr. Kallol Bagchi
Aalborg University
Fredrick Bajers Vej 7
DK-9220 Aalborg Øst
DENMARK
Phone: + 45 98 15 42 11, x4901
Fax: + 45 98 15 67 40
E-mail: kkb@iesd.auc.dk

Dr. Patrick Dowd
SUNY/Buffalo
Bell Hall
Buffalo, NY 14260
USA
+1 716 636-2406
+1 716 636-3656
dowd@eng.buffalo.edu

Coming in future *ConneXions:*

"The Guy with The Bike"
"Prospero: A Virtual Directory Service for the Internet"
"How to Win the Battle of Network Management"
"A Summary of the WORLDWIDEWEB System"
"Comparing Compound Document Architectures"
"Multiprotocol Encapsulation over Frame Relay"
"The OSF Distributed Management Environment (DME)"
"The Internet Gopher"
"OSI Conformance Testing"
Special Issue: Electronic Mail and Directory Services

Announcement and Call for Papers

The Sixth USENIX *System Administration Conference* (LISA) will be held in Long Beach, California, October 19–23, 1992.

The annual LISA conference provides a forum in which system administrators from a variety of sites can meet to share new ideas and experiences. A growing success for the past five years, LISA is the only conference which focuses specifically on the needs of system administrators. In previous years, LISA has targeted large installations. However, this year we are extending the scope of LISA to include system administrators from all UNIX sites.

Format

A dual-track tutorial program will be offered during the first two days of the conference, followed by a three day technical conference. The tutorial program will address issues in introductory and advanced system administration.

Topics

The program committee will be reviewing papers submitted on subjects including (but not limited to):

- Tools for Real-Time System Troubleshooting
- Remote/Off-site System Administration
- Tricks in User Education
- Graphical User Interfaces for System Administration
- Distributed System Administration
- Experiences Using Third-party Administration Software
- Network Growth and Performance Management
- How to Grow Your Own Junior System Administrators
- Network Management
- Wireless LANs
- System Security Monitoring
- Evaluating Performance of High-End Workstations and Servers
- Keys to Successful, Painless Upgrades
- Object Management Systems for System Administration
- Standardization of System Administration
- Heterogeneous System Administration
- System Archiving and Backups

Paper submissions

We are especially interested in papers which provide freely available or fully described solutions to existing problems. We are also looking for papers which, in some way, advance the state of the art.

The committee requires that an extended abstract be submitted for the paper selection process [full-papers are not acceptable for this stage; if you send a full paper, you must also include an extended abstract for judging]. Your extended abstract should consist of a traditional abstract which summarizes the content/ideas of the entire paper, followed by a skeletal outline of the full paper. Final papers should be from 5 to 20 pages in length, including diagrams and figures. Papers should include a brief description of the site, an outline of the problem and issues, and a description of the solution. We require electronic form of the extended abstract; we require both hardcopy and electronic (*nroff*/*troff* or ASCII) form of the final paper.

Proceedings

Conference proceedings will be distributed to all the attendees and also will be available after the conference from the USENIX Association.

BOFs, etc.

In addition to tutorials and regular technical sessions, a handful of other events will be included as part of the program. For example, the program may include special panels, work-in-progress reports, birds-of-a-feather (BOF) sessions, and invited talks. The program committee invites you to submit informal proposals, ideas, or suggestions you might have on any of these topics.

Important dates

Extended Abstract Deadline:	June 29, 1992
Acceptance Notification:	July 20, 1992
Final Papers Received:	August 31, 1992

Contact information

Submit electronic copy of extended abstracts (preferably by electronic mail) to:

Trent Hein
XOR Computer Systems
2525 Arapahoe, Suite E4-264
Boulder, Colorado 80302
Phone: 303- 440-6093
E-mail: `trent@xor.com`

Program Committee

Trent Hein (Program Chair)	XOR Computer Systems
Rik Farrow	*UNIX World*
Jeff Forys	University of Utah
John Hardt	Martin Marietta Astronautics
Rob Kolstad (Board Liaison)	Berkeley Software Design, Inc.
Herb Morreale	XOR Computer Systems
Pat Parseghian	AT&T Bell Laboratories
Jeff Polk	Sun Microsystems

Call For Anthology Papers

I am seeking papers for a book on networking in less developed nations. The tentative title is "Toward a Truly Global Network."

Topics

The anthology will be broad, encompassing descriptions of networks, technology, applications, and social, economic and political considerations and implications. It will include papers on what is currently happening and visions of the future. The primary intended readers are people now working in the area and those who might like to begin doing so. An appendix will refer readers to resources and projects, and it will be assumed that authors and readers are on the Net. Previously published, revised or original papers are suitable.

Submitting papers

If you are interested in publishing a paper in this anthology, please let me know, sending your tentative title, an abstract or short outline, and an estimate of the length. If the paper has been published previously, please send the full text. Send your titles and/or papers to:

Professor Larry Press
CSUDH
10726 Esther Avenue
Los Angeles, CA 90064
Phone: +1 310-475-6515
Fax: +1 310-516-3664
Internet: `lpress@venera.isi.edu`

Call for Participation

The *Third International Symposium on Integrated Network Management* (ISINM '93) will be held April 4–9, 1993 at the historic Sheraton Palace Hotel in San Francisco, California.

Sponsors

Subtitled "Strategies for the Nineties," The symposium is sponsored by the International Federation for Information Processing (IFIP) Working Group 6.6 on Network Management for Communication Networks, with participation by the IEEE Communications Society Technical Committee on Network Operations and Management (CNOM) and support from the Institute for Educational Services (IES).

Focus

The Third International Symposium on Integrated Network Management will build on the success of ISINM '89 and ISINM '91 in forming a central technical forum for the research, standards, development, systems integrator, vendor and user communities of network management. The second symposium demonstrated an increasing interest in overall management solutions across all types of networks, enterprise communication systems, telecommunications, distributed computing systems and applications. Such comprehensive management is thus the focus of the third symposium. This focus includes all aspects of the network, integrating data and telecommunications—from narrowband to broadband, terrestrial to satellite, and stationary to mobile used for ordinary as well as advanced multi-media communications.

Topics

Authors are invited to submit unpublished papers, as well as proposals for tutorial and panel discussions in the following topic areas:

- Standards, Layer Management Approaches, OSI, TMN, Internet, etc.
- Models and Architecture
- Fault, Performance, Configuration, Security & Accounting Mgmt.
- Management Protocols and Protocol Management
- Interoperability and Cooperative Operation
- Telecommunications Management including OAM&P
- Quality of Service Management
- Broadband and Mobile Communications Impacts
- User Requirements and Analysis for Integrated Systems Mgmt.
- Case Study Experiences: Solutions, Limitations and Challenges
- Users' Perspectives and Needs
- Modeling & Storage: Database Techniques, Object Oriented Mgmt.
- Information Interpretation: AI Techniques, Rule-Based Mgmt.
- Neural-Networks Modeling
- Interplay of Distributed Systems and Telecommunications Mgmt.
- Application and Distributed Systems Management
- Formal Methods

Submissions

Please submit seven (7) copies of complete papers in English to either address listed below. The cover page must contain the paper title, brief abstract, list of keywords, author(s) full name(s), affiliation(s), complete address(es), telephone number(s), and (optionally) electronic mail address(es). All submissions will be carefully reviewed and returned to the authors with comments by renowned experts and the Program Committee to ensure high quality. The authors of accepted papers will receive suggested modifications for inclusion in the widely distributed hardbound Symposium Proceedings.

Send complete papers, as well as proposals for tutorials and panel discussions to either:

Yechiam Yemini
Program Co-Chair (Americas, Australia)
Columbia University
450 Computer Science Building
New York, NY 10027
USA

or:

Heinz-Gerd Hegering
Program Co-Chair (Europe, Asia, Africa)
University of Munich
Institut für Informatik—LRZ
Barer Straße 21
D-8000 München 2
GERMANY

Important dates

Deadline for receipt of papers:	July 1, 1992
Deadline for receipt of proposals for tutorials and panel discussions:	July 1, 1992
Notification of Acceptance mailed:	November 1, 1992
Final camera-ready papers due:	December 1, 1992

Vendor Program

The Third International Symposium on Integrated Network Management offers vendors the opportunity to demonstrate their management products for the integrated network management community. Please call +1 415-512-1316 for more information on how your company can participate in the ISINM '93 Vendor Program.

More information

For further information about the symposium contact:

ISINM '93
P.O. Box 191885
San Francisco, CA 94119-1885
Phone: +1 415-512-1316
Fax: +1 415-512-1325
E-mail: `4367585@mcimail.com`

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

ADDRESS CORRECTION
REQUESTED